

"ESTUDO DE ALGORITMOS DE BUSCA EM GRAFOS  
E SUA APLICAÇÃO A PROBLEMAS DE PLANEJAMENTO"

CLOVIS CAESAR GONZAGA

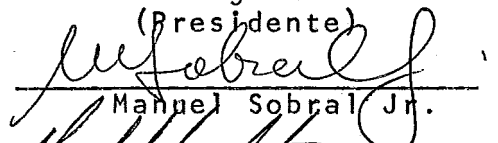
TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE  
PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO  
GRAU DE DOUTOR EM CIÊNCIA (D.Sc.).

Aprovada por:

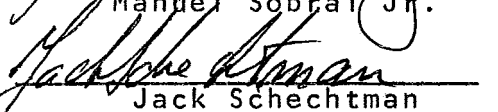
- - - - -



Nelson Ortigosa da Cunha  
(Presidente)



Manuel Sobral Jr.



Jack Schechtman



Ysmar Vianna e Silva F.



Antonio Salles Campos F.

RIO DE JANEIRO  
ESTADO DA GUANABARA - BRASIL  
OUTUBRO DE 1973

À Tânia

### AGRADECIMENTOS

Desejo agradecer ao Professor Nelson Ortégosa da Cunha, não somente pela orientação, mas principalmente pela amizade com que acompanhou meu trabalho.

A Sérgio S. Brito devo o tema da pesquisa e as constantes discussões relacionadas à sua aplicação, um trabalho de equipe realizado em colaboração com Ronaldo C. Marinho Persiano e com o eficiente analista Osias Appel.

### BIOGRAFIA DO AUTOR

Clóvis Caesar Gonzaga nasceu a 6 de setembro de 1944 em Lajes, estado de Santa Catarina. Permaneceu nesse estado até ingressar no Instituto Tecnológico da Aeronáutica, em São José dos Campos, São Paulo, onde recebeu em 1967 o título de Engenheiro de Eletrônica. Ingressou na COPPE no ano seguinte, completando o mestrado em Engenharia Elétrica em 1970. Faz parte do corpo docente da COPPE desde 1969, ensinando e pesquisando em áreas relacionadas a otimização e controle.



## RESUMO

Estudam-se técnicas de Programação Heurística para a busca de caminhos de custo mínimo em grafos. Um algoritmo bastante geral é proposto e estuda-se sua extensão a grafos dotados de uma ordenação parcial do conjunto de nós. A aplicação dessas técnicas a problemas de decisões sequenciais conduz a um método para a resolução de problemas de planejamento a longo prazo de sistemas descritos por redes. Esses problemas são formalizados, fazendo-se um estudo detalhado da otimização a curto prazo, necessário à resolução do problema de otimização a longo prazo. Esses resultados são particularizados para o caso de redes de transmissão de energia elétrica, que é completamente resolvido, apresentando-se finalmente resultados relativos ao planejamento de dois sistemas reais.

## ABSTRACT

Heuristic Programming techniques are studied, related to the search for minimum-cost paths in a graph. A rather general algorithm is proposed, being afterwards extended to graphs in which a partial ordering is defined for the set of nodes. The application of these techniques to sequential decision problems leads to a method for the solution of long term planning problems related to systems modelled by networks. These last problems are formalized and short term optimization is closely studied, in order to be used by the long term optimization schemes. These results are then particularized to the case of electric power transmission networks, which is completely solved. The results obtained by the long term optimization of two real power systems are finally presented.

## INDICE

CAPÍTULO I - INTRODUÇÃO .....	1
CAPÍTULO II - PROBLEMAS DE PLANEJAMENTO .....	7
Seção 1 - O Modelo .....	9
Seção 2 - Otimização a Curto Prazo .....	15
Seção 3 - Otimização a Longo Prazo .....	22
CAPÍTULO III - OTIMIZAÇÃO A CURTO PRAZO .....	27
Seção 1 - O Problema de Expansão .....	29
Seção 2 - Resolução do Problema de Expansão Generalizado	40
CAPÍTULO IV - MODELAGEM E EXPANSÃO DE SISTEMAS DE TRANSMISSÃO DE POTÊNCIA .....	50
Seção 1 - O Modelo do Sistema .....	52
Seção 2 - Análise de contingências .....	59
Seção 3 - Cálculo das Perdas .....	63
Seção 4 - Expansão de Redes de Transmissão de Energia E- létrica .....	64
CAPÍTULO V - PROGRAMAÇÃO HEURÍSTICA E PROCESSOS DE DECISÕES SE- QUENCIAIS .....	74
Seção 1 - Expansão do Algoritmo A* .....	76
Seção 2 - Otimização em Grafos dotados de uma Relação de preferência .....	88
Seção 3 - Grafos em Camadas e Problemas de Decisão Sequen- cial .....	100

CAPÍTULO VI - OTIMIZAÇÃO A LONGO PRAZO .....	109
Seção 1 - Resolução de PLP .....	110
Seção 2 - Otimização a Longo Prazo de Redes de Trans- missão de Potência .....	121
CAPÍTULO VII - IMPLEMENTAÇÃO DO MÉTODO .....	125
Seção 1 - O Programa .....	126
Seção 2 - Estudo de Casos .....	129
CAPÍTULO VIII - CONCLUSÃO .....	140
APÊNDICE A - TEORIA DE GRAFOS .....	142
APÊNDICE B - O ALGORITMO A* .....	149
REFERÊNCIAS .....	154

## CAPÍTULO I

### INTRODUÇÃO

O presente trabalho originou-se em um problema real ligado à economia brasileira. Trata-se do planejamento da expansão a longo prazo do sistema de transmissão de potência da região centro sul do Brasil. A grande extensão territorial da região e a elevada taxa de crescimento da demanda no sistema tornam excepcionalmente rentável a aplicação de técnicas elaboradas de **otimização**, em contraposição às sistemáticas tradicionais de planejamento, baseadas na intuição de pessoal experimentado.

Embora a pesquisa seja suscitada pelo estudo de sistemas elétricos, não se restringe a esses sistemas: estudam-se técnicas gerais para a resolução de problemas de decisões sequenciais, aplicados posteriormente a sistemas de transmissão de potência.

O estudo é feito em três níveis, de generalidade decrescente. No primeiro nível, propõe-se uma abordagem geral a problemas de decisão sequencial discretos, formulando-os em termos de grafos e propondo-se um método para sua solução baseado em **Programação Heurística** em grafos. Esse estudo consta do Capítulo V e independe do restante da tese.

No segundo nível, propõe-se uma **formalização** bastante geral aos problemas de planejamento da expansão de sistemas descritos por redes (capítulo II). Um método para a solução de problemas de planejamento a curto prazo é apresentado no Capítulo III, estu -

I.

dando-se o planejamento a longo prazo no Capítulo VI. O restante da tese ocupa-se do **nível** de menor generalidade, aplicando as **téc**nicas propostas a sistemas **elétricos**, finalizando pela apresentação de alguns exemplos práticos de sua aplicação ao planejamento da expansão de vários sistemas reais de transmissão de potência.

Neste capítulo, comenta-se informalmente o conteúdo da tese, segundo os níveis de generalidade apontados acima.

Problemas de Decisão Sequencial: Entre os **possíveis** problemas de decisão sequencial, tem importância fundamental os problemas **discretos determinísticos** com espaço de estado finito, devido aos aspectos computacionais de seu tratamento. Sua resolução tem sido abordada principalmente sob o enfoque de Programação Dinâmica [1], encontrando-se sérias dificuldades relacionadas a requisitos computacionais à medida que cresce o porte dos sistemas considerados.

Pouca ênfase tem sido dada ao fato de que, para esses problemas, a Programação Dinâmica pode ser considerada como um método de busca em grafos (Ver [2]), sendo sua eficiência bastante pobre quando comparada, por exemplo, ao **algoritmo** de **Dijkstra** [3], como se comenta em [4]. A aplicação de técnicas eficientes de busca em grafos em **pesquisa** operacional é extensa, no contexto de métodos de "Branch and Bound" [5], aplicados à Programação Inteira. Não parece existir, no entanto, nenhum esforço de unificação desses métodos no que diz respeito à sua aplicação a problemas de decisão sequencial. Como resultado, pesquisadores não especializados em grafos ou programação inteira tendem a limitar-se a aplicações de Programação Dinâmica. As técnicas e a terminologia de gra

1.

fos tem-se limitado a especialistas, ou a campos como ciência de **computação** e inteligência artificial.

Uma nova abordagem a problemas de otimização consiste na utilização de informações adicionais porventura existentes a respeito de cada problema particular, resultando em um campo bem formalizado em [6], a Programação Heurística. Originada no contexto de Inteligência Artificial, a Programação Heurística em grafos tem-se conservado restrita a esse campo, sendo estudada em [2], [7], [8]

Neste trabalho, propõe-se uma extensão ao algoritmo A" desenvolvido em [6], de modo a adaptá-lo à resolução de problemas de decisão sequencial envolvendo sistemas de grande porte. Desenvolve-se também uma técnica especial para o tratamento de grafos cujos nós admitem uma ordenação parcial capaz de acelerar a operação do algoritmo.

Problemas de Planejamento : Um grande **número** de sistemas de grande porte admite uma modelagem por redes **finitas** de maneira imediatamente aparente. O planejamento da evolução de tais sistemas pode prender-se a características dos ramos ou dos **nós** das redes que os **modelam**. Neste trabalho, a ênfase **é** colocada no planejamento de certas **características** dos ramos, descritas em geral como o "estado" dos ramos.

Exemplos **típicos** de sistemas aos quais aplica-se esta abordagem são redes de transmissão de energia elétrica, sistemas de distribuição de **água**, coleta de esgotos, oleodutos, gasodutos, todos eles modelados por redes com fluxo **não** endereçado. Sistemas de tráfego aéreo, terrestre ou marítimo e sistemas de transmissão de

I.

informação apresentam o problema adicional de fluxos endereçados, o que dificulta sua modelagem. Sistemas de reservatórios de água interligados exigem o planejamento de características dos nós, o que também ocorre em sistemas de geração de energia elétrica: esses problemas não podem ser tratados imediatamente pelo método proposto, sendo necessários desenvolvimentos adicionais.

O problema a ser estudado consiste, portanto, no seguinte: um sistema descrito por uma rede finita com fluxo limitado por parâmetros associados a seus nós e ramos deve evoluir a partir de uma configuração inicial conhecida, durante um número finito de estágios de planejamento. A evolução do sistema consiste na modificação do estado de seus ramos de modo a satisfazer condições de viabilidade dependentes do estado. Procura-se uma sequência ótima de transições de estado nos ramos da rede, de modo que as condições de viabilidade sejam satisfeitas a cada estágio e que o custo total da estratégia (levando em conta implementações e custos de operação), seja mínimo.

Tomando como exemplo um sistema de transmissão de potência, procura-se uma estratégia de construções de linhas de transmissão de mínimo custo entre as estratégias que garantam a estabilidade e segurança do sistema a cada estágio, conhecida a programação de produção e demanda de energia:

Planejamento a Curto Prazo : No planejamento a curto prazo, considera-se somente um estágio, o que simplifica radicalmente o problema. Trata-se então de um problema de programação matemática, que consiste em encontrar um estado para o sistema que o viabilize a mínimo custo. Esse problema tem sido resolvido por várias técnicas a



1.

proximadas para sistemas de transmissão de potência, como métodos gradientes [9], programação linear aplicada a uma aproximação contínua [10], análise de alternativas usando informações heurísticas baseadas na experiência de planejadores [11], ou ainda métodos discretos simplificados para viabilização, sem preocupação com otimalidade [12].

Neste trabalho, desenvolve-se uma técnica geral para a resolução de uma generalização do problema de otimização a curto prazo, utilizando um algoritmo de busca em grafos capaz de minimizar a memória de computador ocupada (Ver Capítulo III).

Planejamento a Longo Prazo : Trata-se de um problema de decisões sequenciais, cuja resolução tem sido tentada por meio de Programação Dinâmica [10], [12]. A otimização a curto prazo deve ser realizada a cada estágio, fornecendo as transições de estado necessárias ã solução dos problemas de decisão sequencial.

Uma simples sequência de otimizações a curto prazo gera uma solução rápida, chamada solução incremental, que em alguns casos fornece bons resultados, como se comenta em [13], com relação ã ex-pansão da geração de energia elétrica. Em geral, resultados superiores são obtidos por uma otimização global, como se exemplifica no Capítulo VII .

Como o problema de que se estã tratando costuma ter grande porte, torna-se inviãvel uma aplicação direta de Programação Dinãmica, o que motivou o desenvolvimento de técnicas baseadas em aprimoramentos iterativos de soluções iniciais do tipo incremental [10] , [12], ou em modificações do algoritmo de Programação Dinâmica para introduzir certas características típicas a algoritmos de busca em grafos [14], ou em procedimentos não determinísticos [15]. Uma modi

1.

**ficação** do algoritmo de Dijkstra aplicada a problemas de planejamento a longo prazo foi desenvolvida em [11].

A abordagem seguida neste trabalho baseia-se nas técnicas de resolução de problemas de decisão sequencial desenvolvidas no Capítulo V, e procura diretamente uma sequência de **soluções sub-ótimas** para o problema de planejamento. Essa sequência tende em tempo finito para uma solução Ótima, terminando o processo, ou **pára** ao esgotar-se o tempo de computação disponível, fornecendo uma **solução sub-ótima** nunca pior do que a estratégia incremental (Ver Capítulo VI).

**Expansão de Sistemas de Transmissão de Potência** : Desenvolve-se no Capítulo IV o modelo de desempenho de redes de transmissão empregado em estudos de planejamento, que se baseia em uma aproximação linear dos testes de Fluxo de Carga. Um estudo de sensibilidade fornece um método para a execução **rápida** de análises de contingências.

Com essas ferramentas, descreve-se um método para a **otimização** a curto prazo, utilizando o **algoritmo comentado** acima ( Ver **Capítulo IV**). A **otimização** a longo prazo resulta da aplicação direta dos **resultados** gerais comentados acima (Ver **Capítulo VI**).

Comenta-se, finalmente, no Capítulo VII, uma **implementação** do método desenvolvido, apresentando-se alguns exemplos de projetos ligados a sistemas de transmissão de potência brasileiros.

## CAPÍTULO II

### PROBLEMAS DE PLANEJAMENTO

Neste capítulo expõem-se formalmente os problemas de planejamento propostos. O formato escolhido para a formalização é suficientemente geral para englobar problemas pertencentes a vários campos de estudo (Ver Seção 4), mas evita-se uma generalização excessiva. A motivação para a formalização proposta está em sistemas de transmissão de energia elétrica, o que ficará evidente nos capítulos IV e VII, onde se trata especificamente desses sistemas. Procura-se portanto, manter a **generalidade** do problema formal compatível com a complexidade dos sistemas de transmissão que se deseja estudar. Com essa motivação, procura-se planejar somente características dos ramos das redes que modelam os sistemas (linhas de transmissão). As características dos nós (geração de potência) são consideradas conhecidas a priori, deterministicamente. No entanto, para permitir o emprego do modelo ao estudo da geração de potência, ou ao planejamento de reservatórios de água, incluem-se na seção 4 alguns **possíveis** caminhos para a generalização ou particularização do modelo.

Utiliza-se extensivamente, teoria de grafos em todo o trabalho e torna-se conveniente listar alguns conceitos e resultados dessa teoria, o que é feito no Apêndice A. Tratamentos extensos encontram-se em [16] , [17] , [18] .

11.

Pretende-se planejar a evolução da estrutura de um sistema. O estudo parte de uma estrutura **básica** geral do tipo rede finita, limitando-se o sistema a assumir estruturas dadas por **sub-re**-des parciais da rede básica. Define-se então configuração como cada uma dessas sub-redes, dotada de um estado que caracteriza a composição de seus ramos. Enunciam-se a seguir os problemas de **otimi**-zação a curto e a longo prazo.

## 11.1

Seção 1 - O MODELO

1 Definição: Seja  $(N, M)$  uma rede. Diz-se que uma rede  $(N', M')$  es-  
tá contida em  $(N, M)$  ,

$(N', M') \subset (N, M)$  , se  $(N', M')$  é uma sub-rede parcial de  $(N, M)$ .

2 Considere-se uma rede finita  $(\bar{N}, \bar{M})$ , chamada rede básica, onde:

$\bar{N} = (\bar{n}_j)_{j=1,2,\dots,\bar{n}}$  , é o conjunto de nós básicos,  $|\bar{N}| = \bar{n}$

$\bar{M} = (\bar{r}_i)_{i=1,2,\dots,\bar{m}}$  , é a família de ramos básicos,  $|\bar{M}| = \bar{m}$  .

Tem-se portanto (Ver A.1),  $\bar{r}_i \in \bar{N} \times \bar{N}$  e permitem-se ramos repetidos, uma vez que  $\bar{M}$  é uma família.

3 Convenções - Seja  $(\bar{N}, \bar{M})$  uma rede básica (2) e considere-se uma rede  $(N, M) \subset (\bar{N}, \bar{M})$ , com  $n$  nós e  $m$  ramos. Fazem-se as seguintes convenções:

$N = (n_q)_{q=1,2,\dots,n}$  , onde  $n_q = \bar{n}_{i_q}$  ,  $q=1,2,\dots,n$

$M = (r_p)_{p=1,2,\dots,m}$  , onde  $r_p = \bar{r}_{i_p}$  ,  $p=1,2,\dots,m$

Dada uma família qualquer

$(\bar{\alpha}_i)_{i=1,2,\dots,\bar{m}}$  , indexada segundo os ramos de  $(\bar{N}, \bar{M})$ , deno-  
ta-se por  $(\alpha_i)_{i=1,2,\dots,m}$  a sub-família correspondente aos ramos  
de  $(N, M)$ . O mesmo tipo de reindexação é feita para famílias inde-  
xadas segundo os nós de  $(\bar{N}, \bar{M})$  e  $(N, M)$ .

## 11.1

4 Considerem-se os conjuntos finitos

$\bar{\sigma}_i \subset R$ ,  $i=1,2,\dots,m$ , associados aos ramos de uma rede básica  $(\bar{N},\bar{M})$ . Chama-se a cada  $\bar{\sigma}_i$  de conjunto de estados admissíveis para o ramo  $\bar{r}_i$ . Seguindo as convenções (3), associam-se conjuntos  $\sigma_p = \bar{\sigma}_{i_p}$ ,  $p=1,2,\dots,m$  aos ramos de uma rede contida em  $(\bar{N},\bar{M})$ .

5 Definição: Uma configuração da rede básica  $(\bar{N},\bar{M})$  (ou simplesmente configuração, se não houver dúvidas quanto a  $(\bar{N},\bar{M})$ ), é uma tripla ordenada  $C = (N,M,S)$ , onde

(i)  $(N,M) \subset (\bar{N},\bar{M})$  é uma rede, chamada topologia da configuração C

(ii)  $S \in R^m$  é um vetor de estado, cada componente de S satisfaz

$S_i \in \sigma_i$  e é chamada estado do ramo  $r_i$ .

Usam-se as convenções (3).

6 Definição: Dada uma rede  $(N,M) \subset (\bar{N},\bar{M})$ , define-se o espaço de estados admissíveis associado à topologia  $(N,M)$  por

$$\sigma = \sigma_1 \times \sigma_2 \times \dots \times \sigma_m.$$

Observe-se que (5 ii) poderia ser escrito  $S \in \sigma$ .

Pretende-se estudar problemas de planejamento das características dos ramos de sistemas modelados por redes. Uma configuração para um tal sistema é modelada por uma 'configuração' segundo a definição acima, sendo comum definirem-se condições de viabilidade para o sistema baseadas no conceito de capacidade associada aos

## 11.1

ramos da rede que o modela:

Considerem-se dados os mapeamentos

$$7 \quad \bar{\gamma}_i : \bar{\sigma}_i \rightarrow R, \quad i=1,2,\dots,\bar{m}$$

que associam a cada estado admissível  $s \in \bar{\sigma}_i$  para o ramo  $i$  da rede básica, a capacidade  $\bar{\gamma}_i(s)$  do ramo  $i$  no estado  $s$ .

Dada uma configuração  $C = (N, M, S)$ , particularizam-se os mapeamentos  $\gamma_i$  segundo as convenções (3), definindo-se o vetor de capacidades associado a  $C$  por

$$[\gamma(s)]_p = \gamma_p(s_p), \quad p=1,2,\dots,m.$$

A figura 11.1 ilustra uma rede básica e duas possíveis configurações.

8 Matrizes de Incidência - Conhecida a matriz de incidência da rede básica, é imediata a determinação das matrizes de incidência para suas configurações. Seja  $C = (N, M, S)$ , conforme (5), onde

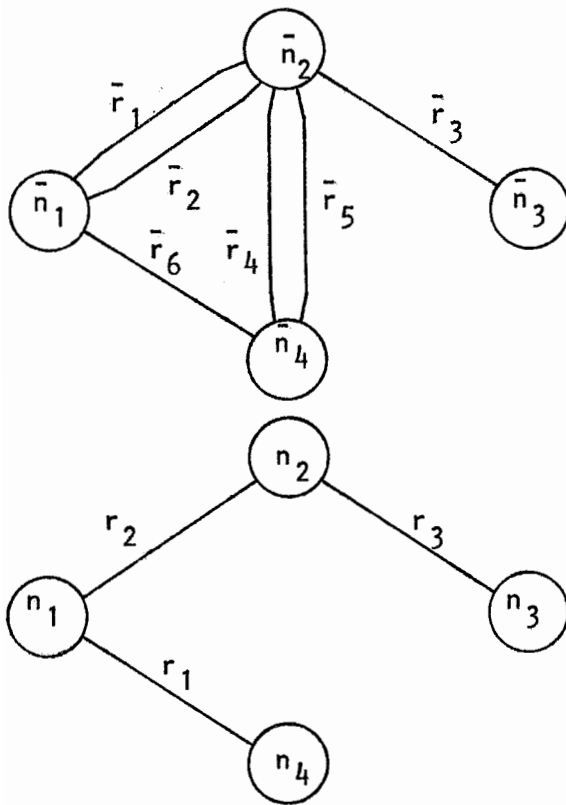
$$N = (n_1, n_2, \dots, n_n) = (\bar{n}_{j_1}, \bar{n}_{j_2}, \dots, \bar{n}_{j_n})$$

$$M = (r_1, r_2, \dots, r_m) = (\bar{r}_{i_1}, \bar{r}_{i_2}, \dots, \bar{r}_{i_m})$$

Seja  $\bar{T}$  a matriz de incidência de  $(\bar{N}, \bar{M})$ . A matriz de incidência  $T$  de  $(N, M)$  é dada por

$$T_{qp} = \bar{T}_{i_q i_p}, \quad q=1,2,\dots,n \quad p=1,2,\dots,m$$

11.1



$$\bar{n} = 4$$

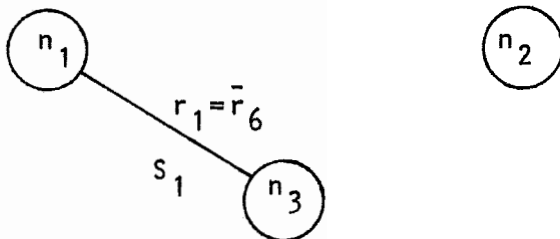
$$\bar{m} = 6$$

$$n_i = \bar{n}_i, \quad N = \bar{N}$$

$$M = (r_1, r_2, r_3) = (\bar{r}_6, \bar{r}_1, \bar{r}_3) = (\bar{r}_{i_1}, \bar{r}_{i_2}, \bar{r}_{i_3})$$

$$i_1 = 6, \quad i_2 = 1, \quad i_3 = 3$$

$$\sigma_1 = \bar{\sigma}_6, \quad \sigma_2 = \bar{\sigma}_1, \quad \sigma_3 = \bar{\sigma}_3$$



$$N = (n_1, n_2, n_3) = (\bar{n}_1, \bar{n}_3, \bar{n}_4)$$

$$M = (r_1 = (\bar{r}_6))$$

$$s_1 \in \sigma_1 = \bar{\sigma}_6, \quad \gamma_1 = \bar{\gamma}_6$$

$$\gamma_1(s_1) = \bar{\gamma}_6(s_1)$$

Fig. 11.1 - Rede básica de dois exemplos de configurações



Exemplo - A matriz de incidência da rede básica da fig. 11.1 é

$$\bar{T} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & -1 \\ -1 & -1 & 1 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 1 \end{pmatrix} \end{matrix}$$

Para a primeira configuração da figura,

$$\bar{T} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{pmatrix} \end{matrix} \quad \begin{matrix} i_q = q \\ i_1 = 6, \quad i_2 = 1, \\ i_3 = 3 \end{matrix}$$

Se os sistemas em estudo são modelados por redes com fluxo, deve-se considerar a ocorrência de fontes e sumidouros. Em geral, consideram-se parâmetros associados aos nós de uma rede assumindo-se que o sistema fica totalmente especificado por sua configuração segundo (5) e pelos parâmetros dos nós.

- 9 Definição - O vetor de parâmetros dos nós é um vetor  $n$ -dimensional  $P \in \mathbb{R}^n$ , cuja componente  $P_j$ ,  $j=1,2,\dots,n$  é chamada parâmetro do nó  $n_j$  da configuração.

## 11.1

- 10 Viabilidade de uma Configuração: Suponha-se que existe uma lei de formação de **vínculos**, que associa a cada configuração um conjunto de vínculos

$$C = (N, M, S) \mapsto g_C$$

onde

$$g_C : R^m \times R^n \rightarrow R^k, \quad k \geq 1, \text{ dependente de } C$$

é tal que

$$\gamma \in R^n, P \in R^m \mapsto g_C(\gamma, P) \in R^k$$

- 11 Definição - Uma configuração  $C = (N, M, S)$  da rede básica  $(\bar{N}, \bar{M})$  é dita **viável** com respeito aos **parâmetros**  $P \in R^n$  se

$$g_C(\gamma(S), P) \leq 0$$

Os vínculos  $g_C$  podem ser **muito** complexos, podendo mesmo ser **vínculos** lógicos, dependentes da resolução de outros problemas de **otimização** (Ver [10] ), ou dependentes de **sequências** de testes executados sobre o sistema modelado. Em geral **é difícil co**locar os **vínculos** no formato (11), pois a viabilidade de  $C$  é **tes**tada por um programa separado em computador. Os sistemas **elétricos** estão neste caso, como se **verá** no Capítulo IV, onde suas **condi**ções de viabilidade são desenvolvidas. Em outros casos, os **víncu**los são simplesmente uma limitação dos fluxos **permissíveis** nos ramos.

## Seção 2 - OTIMIZAÇÃO A CURTO PRAZO

Na otimização a curto prazo, procura-se **viabilizar** uma configuração dada com respeito a um vetor de **parâmetros também** dado, por meio de uma mudança de estado de **mínimo** custo. O custo deve **le**var em conta as características da configuração resultante (custos de operação) e o custo da transição de estado.

Embora a otimização a curto prazo **seja, por** definição, executada em apenas um estágio, deve-se desde **já** introduzir este **fa**tor, uma vez que o objetivo do trabalho encontra-se na otimização a longo prazo. Quando se tratar somente de otimização a curto prazo, basta fazer  $T = \{0\}$  abaixo.

Considere-se então dada uma **família** finita de estágios

$$12 \quad T = (1, 2, \dots, f)$$

Sejam também dados a  $t \in T$ ,  $\forall s \in \bar{\sigma}_i$ ,  $\forall i=1, 2, \dots, \bar{m}$ , os conjuntos

$$13 \quad \delta_i(s, t) \subset \bar{\sigma}_i, \quad \text{denominados conjuntos de estados su}$$

cessores de  $s$  para o ramo  $i$  no estágio  $t$ .

Esses conjuntos limitam as **transições** de estado admitidas em cada **es**tágio. Como de costume (3), pode-se reindexar os conjuntos para uma configuração particular, definindo-se **também** o conjunto de estados sucessores de  $S \in R^m$  no **estágio**  $t$  por

$$\delta(s, t) = \delta_1(s_1, t) \times \delta_2(s_2, t) \times \dots \times \delta_m(s_m, t).$$

Passa-se agora **à** caracterização de configurações sucesoras e **à** definição de expansão de uma configuração, conceitos esses

## 11.2

fundamentais para toda a **sequência**.

Considere-se conhecida uma rede básica  $(\bar{N}, \bar{M})$ .

- 14 Seja  $\bar{C} = (\bar{N}, \bar{M}, \bar{S})$  uma configuração básica dada, isto é, uma configuração cuja topologia é  $(\bar{N}, \bar{M})$ . Sejam dadas uma configuração  $C^0 = (N^0, M^0, S^0)$  e uma topologia  $(N, M) \subset (\bar{N}, \bar{M})$  com

$$N^0 = (n_q^0)_{q=1,2,\dots,n_0} \quad N = (n_q)_{q=1,2,\dots,n}$$

$$M^0 = (r_p^0)_{p=1,2,\dots,m_0} \quad M = (r_p)_{p=1,2,\dots,m}$$

Supondo-se ainda que

$$(N^0, M^0) \subset (N, M) \text{ , com a convenção}$$

$$15 \quad n_q^0 = n_q \quad q=1,2,\dots,n_0$$

$$r_p^0 = r_p \quad p=1,2,\dots,m_0$$

Ou seja, os primeiros nós e ramos coincidem em ambas as topologias.

Conhecidos os elementos introduzidos em (14), sejam dados um vetor de parâmetros  $P \in \mathbb{R}^n$  e um estágio  $t \in T$ .

- 16 Definição - Uma configuração  $C = (N, M, S)$  é sucessora de  $C^0$  com respeito aos parâmetros  $P$  e à topologia  $(N, M)$  no estágio  $t$  (ou simplesmente sucessora quando não houver confusão possível) se

$$(i) \quad (N^0, M^0) \subset (N, M)$$

$$(ii) \quad s_p \in \delta_p(s_p^0, t) \quad p=1,2,\dots,n_0$$

## 11.2

$$s_p \in \delta_p(\bar{s}_{i_p}, t) \quad p=n_0+1, \dots, n \quad (\text{usando os \u00edndices de (15)})$$

(iii)  $C$  \u00e9 vi\u00e1vel com respeito a  $P$ .

**17 Defini\u00e7\u00e3o** - A expans\u00e3o de uma configura\u00e7\u00e3o com respeito a uma topologia  $\tau = (N, M)$  e ao vetor de par\u00e2metros  $P$  no est\u00e1gio  $t \in T$ , consiste na obten\u00e7\u00e3o de todas as sucessoras de  $C^0$  com respeito a  $\tau$  e  $P$  no est\u00e1gio  $t$ . Ao mapeamento  $\Gamma$ , tal que

$$C^0, \tau, P, t \mapsto \Gamma(C^0, \tau, P, t) = \{C \mid C \text{ sucessora de } C^0 \text{ c.r. a } \tau \text{ e } P \text{ no est\u00e1gio } t\}$$

definido para toda configura\u00e7\u00e3o  $C^0 = (N^0, M^0, S^0)$  de  $(\bar{N}, \bar{M})$ , para toda topologia satisfazendo  $(N^0, M^0) \subset \tau$ , para todo vetor de par\u00e2metros  $P$  compat\u00edvel com  $\tau$ ,  $t \in T$ , chama-se operador sucessor de  $(\bar{N}, \bar{M})$ .

**18 Observa\u00e7\u00e3o** - Como todos os conjuntos  $\bar{\sigma}_i$  s\u00e3o finitos,  $\Gamma(C^0, \tau, P, t)$  ser\u00e1 sempre um conjunto finito.

Ao estudar problemas de expans\u00e3o de sistemas descritos por redes, \u00e9 interessante deter-se sobre as poss\u00edveis transi\u00e7\u00f5es de estados em tais sistemas. Encontram-se usualmente algumas transi\u00e7\u00f5es de estado elementares, que consistem na pr\u00e1tica implementa\u00e7\u00e3o de instala\u00e7\u00f5es padronizadas. Definem-se a seguir essas transi\u00e7\u00f5es, sem que se imponha nenhuma nova restri\u00e7\u00e3o, uma vez que podem-se definir como elementares todas as transi\u00e7\u00f5es poss\u00edveis.

## 11.2

- 19 Definição - Considerem-se os conjuntos  $\bar{a}$  introduzidos em (4). Uma transição de estado para o ramo  $i$  é um par ordenado

$$(s, s') \in \bar{\sigma}_i \times \bar{\sigma}_i$$

Uma transição de estado correspondente a uma topologia  $(N, M)$  é um par  $(s, s') \in a \times \sigma$ , com  $a$  definido em (6).

- 20 Considera-se conhecido para cada  $i$ ,  $\bar{\sigma}_i$  um conjunto de transições de estado elementares contido em  $\bar{\sigma}_i \times \bar{\sigma}_i$ , satisfazendo a seguinte condição:

Se  $s, s' \in \bar{a}_i$ , então existe uma sequência finita de estados

$$(s_k)_{k=0,1,2,\dots,K} \quad K \geq 1, \text{ tal que}$$

$$s_0 = s, \quad s_K = s' \quad \text{e}$$

$(s_{k-1}, s_k)$ ,  $k=1, 2, \dots, K$  são transições elementares de estado para o ramo  $i$ .

- 21 Definição - Uma transição de estado  $(s, s')$  correspondente a uma topologia  $(N, M)$  é elementar se existir  $k \leq m$  tal que

$$s_p = s'_p \quad \text{para } p \neq k$$

$$(s_k, s'_k) \text{ é uma transição elementar para o ramo } k.$$

A partir das definições acima, podem-se definir os custos associados às transições de estado na rede.

## 11.2

- 22 Custo da transição de estados: suponham-se conhecidos os seguintes **mapeamentos** correspondentes aos **ramos** de  $(\bar{N}, \bar{M})$  :

$$\bar{c}_i : \bar{\sigma}_i \times \bar{\sigma}_i \times T \rightarrow \mathbb{R} \quad i=1,2,\dots,\bar{m}$$

que associam à mudança de estado em um ramo, um custo atualizado

$s, s' \in \bar{\sigma}_i$  ,  $t \in T \mapsto \bar{c}_i(s, s', t) \in \mathbb{R}$  , satisfazendo as condições de consistência abaixo:

23 (i)  $(\forall \hat{s} \in \bar{\sigma}_i) \quad \bar{c}_i(s, s', t) \leq \bar{c}_i(s, \hat{s}, t) + \bar{c}_i(\hat{s}, s', t)$

(ii) Existe uma **sequência** de transições elementares

$((s_{k-1}, s_k))_{k=1,2,\dots,K}$  ,  $K \geq 1$  , tal que  $s_0 = s$  ,  $s_K = s'$  , satisfazendo

24 
$$\sum_{k=1}^K \bar{c}_i(s_{k-1}, s_k, t) = \bar{c}_i(s, s', t) .$$

Das condições acima, conclui-se imediatamente que o custo de uma transição de estados é dado pelo mínimo entre as **somatórias** (24) para todas as **sequências** de transições conduzindo de  $s$  a  $s'$  .

Esses custos correspondem aos custos **necessários à implementação** de uma mudança de estado em cada estágio e supõe-se que uma implementação será sempre feita a **mínimo** custo.

Pode-se agora definir o custo da transição de uma configuração para outra. Sejam  $\bar{C} = (\bar{N}, \bar{M}, \bar{S})$  ,  $c^0 = (N^0, M^0, S^0)$  ,  $C = (N, M, S)$  configurações definidas conforme (14), isto é, tais que os primeiros **nós** e ramos de  $C$  correspondem aos **nós** e ramos de

## 11.2

$c^0$ . Seja  $t \in T$  um estágio dado.

- 25 Definição : O custo da transição de  $c^0$  para  $C$  no estágio  $t$  é dado por

$$c_t(c^0, C, t) = \sum_{p=1}^{n_0} c_p(s_p^0, s_p, t) + \sum_{p=n_0+1}^n c_p(\bar{s}_{i_p}, s_p, t),$$

onde  $c_p$  são os custos de transição de estados (22). Condições equivalentes a (23) são obtidas imediatamente como **consequência** da definição de  $c_t$  e de (23): existe sempre uma **sequência** de **configurações** entre  $c^0$  e  $C$  correspondentes a transições elementares de estado que fornecem o custo da transição de  $c^0$  para  $C$ .

- 26 Custo de operações : suponha-se conhecido o mapeamento

$$c_o : C, P, t \mapsto c_o(C, P, t) \in \mathbb{R}$$

que associa a cada configuração  $C$  e a qualquer vetor de parâmetros  $P$ , para os **nós** de  $C$  um custo descontado real. Esse custo corresponde à operação do sistema durante um **período pré-determinado**. Tipicamente, corresponde a custos de manutenção de equipamento e perdas de energia.

Pode-se agora passar ao enunciado do problema de **otimização** a curto prazo (**PCP**).

- 27 Problema de otimização a curto prazo (PCP) - Considerem-se conhecidos os elementos introduzidos em (14), ou seja,  $\bar{c}$ ,  $c^0$ ,  $(N, M)$ . Sejam também fornecidos um estágio  $t \in T$  e os parâmetros  $P \in \mathbb{R}^n$ .

PCP : Encontrar uma configuração  $\bar{c}$  sucessora de  $c^0$  com respei-



## 11.2

to à topologia  $(N,M)$  no estágio  $t$  que minimize entre as sucessoras de  $C^0$  com respeito a  $(N,M),t$ , o custo

$$28 \quad c(C^0, C, P, t) \triangleq c_t(C^0, C, t) + c_o(C, P, t)$$

Utilizando o operador sucessor (17), pode-se reescrever :

29 PCP :

$$\begin{aligned} & \dots \quad \text{minimizar} \quad c(C^0, C, P, t) \\ & C \in \Gamma(C^0, (N, M), P, t) \end{aligned}$$

Deve-se lembrar que está implícita em **PCP** a viabilidade da configuração ótima  $\hat{C}$ , uma vez que deve ser sucessora de  $C^0$  (Ver 16) . Note-se também que devido a este fato, o estado de  $\hat{C}$  é limitado pelas transições **admissíveis** no estágio  $t$ .

O problema **PCP** é um problema de programação inteira, uma vez que os conjuntos  $\bar{\sigma}_i$  são finitos por hipótese. Soluções para o caso de sistemas de transmissão de potência foram propostas em [ 9 ] , [ 10 ] , [ 11 ] , [ 14 ] , e serão comentadas no Capítulo IV, onde será apresentado um novo método de solução. Um método de solução para o caso de reservatórios de água foi proposto em [ 20 ] .

## 11.3

Seção 3 - OTIMIZAÇÃO A LONGO PRAZO

A otimização a longo prazo parte de uma configuração inicial, que deve evoluir segundo um número finito de estágios de planejamento. Supõem-se conhecidas as topologias e os vetores de parâmetros dos nós a cada estágio e procura-se otimizar a evolução do estado das configurações restrito pelas condições de viabilidade a cada estágio. Os custos dependem do estágio, levando em conta descontos e condições terminais. Uma análise econômica desses fatores encontra-se em [24] , [25] . Resultados para sistemas elétricos serão apresentados no Capítulo VII.

Considerem-se conhecidos os seguintes elementos:

- 30 (i) Uma família de estágios  $T = (1, 2, \dots, f)$  ,  
 31 (ii) Uma rede básica  $(\bar{N}, \bar{M})$  , conforme (2) ,  
 32 (iii) Um estado inicial  $\bar{S}$  admissível para a rede básica (e portanto, uma configuração básica  $(\bar{N}, \bar{M}, \bar{S})$ ) ,  
 33 (iv) Uma sequência de topologias de configurações de  $(\bar{N}, \bar{M})$  ,

$$((N^0, M^0), (N^1, M^1), \dots, (N^f, M^f)) \quad , \text{ tais que}$$

$$(N^{t-1}, M^{t-1}) \subset (N^t, M^t) \quad \forall t \in T \text{ e tais que para } t=0, 1, \dots, f \quad ,$$

$$|N^t| = n_t \quad , \quad |M^t| = m_t$$

$$N^t = (\bar{n}_{i_1}, \bar{n}_{i_2}, \dots, \bar{n}_{i_{n_t}}) \quad , \quad M^t = (\bar{r}_{i_1}, \bar{r}_{i_2}, \dots, \bar{r}_{i_{m_t}})$$

Note-se que com esta convenção, os primeiros nós e ramos de uma topologia  $(N^t, M^t)$  coincidem com os nós e ramos de  $(N^{t-1}, M^{t-1})$  , do mesmo modo que em (14), o que nos permite utilizar os conceitos definidos anteriormente, relacionados a custos ,

## 11.3

expansão e **otimização** a curto prazo.

- 34 (v) Uma **sequência** de **parâmetros** dos **nós** para as redes correspondentes em (30)

$$(P^1, P^2, \dots, P^f)$$

- 35 (vi) Uma função custo da transição de estados correspondente a cada ramo do  $(\bar{N}, \bar{M})$

$$c_i : \bar{\sigma}_i \times \bar{\sigma}_i \times T \rightarrow R \quad i=1, 2, \dots, \bar{m}, \text{ definida em (22)}$$

- 36 (vii) Uma função custo da operação

$$c_o : C, P, t \mapsto c_o(C, P, t) \in R, \text{ definida em (26)},$$

$c_o$  representa o custo de operação de  $C$  com os parâmetros  $P$  entre os estágios  $t$  e  $t+1$ , se  $t < f$ , ou um custo convencionado para a operação **após** o estágio final  $f$ .

- 37 Definição - Conhecidos os **elementos** (30)-(34), define-se estratégia de expansão ou política de expansão como uma **sequência** de configurações  $E = (C^0, C^1, \dots, C^f)$  tais que

$$(i) \quad C^0 = (N^0, M^0, S^0), \text{ onde } S_p^0 = \bar{S}_{i_p}$$

$$(ii) \quad C^t \text{ é sucessora de } C^{t-1} \text{ com respeito aos parâmetros } P^t \text{ e à topologia } (N^t, M^t) \text{ no estágio } t, \text{ para } t=1, 2, \dots, f.$$

- 38 Definição - Conhecidos os elementos (30)-(36), seja dada uma estratégia de expansão  $E = (C^0, C^1, \dots, C^f)$ . Define-se o custo da estratégia  $E$  por

$$c(E) = \sum_{k=1}^f c_t(C^{k-1}, C^k, k) + c_o(C^k, P^k, k)$$

## 11.3

onde  $c_t$  é o custo da transição definido em (25) .

Pode-se finalmente, definir o problema de **otimização a longo prazo** ,

Problema de Otimização a Longo Prazo (PLP) - Sejam dados uma **família** de **estágios**  $T$  (30), uma configuração **básica**  $(\bar{N}, \bar{M}, \bar{S})$  (32) , uma sequência de topologias (33), uma **sequência** de vetores de **parâmetros** (34) e uma função custo para estratégias de expansão (38) .

39. PLP - Encontrar uma estratégia de expansão

$$\hat{E} = (c^0, \hat{c}^1, \hat{c}^2, \dots, \hat{c}^f)$$

de mínimo custo entre todas as estratégias de expansão **possíveis** com os dados acima.

PLP é um problema de decisões sequenciais, cuja solução tem sido tentada através de programação dinâmica [10] , [14] , [15] para sistemas de transmissão de energia elétrica. Essas soluções serão comentadas no Capítulo VII. Neste **capítulo** propõe-se um **método** de solução para o Problema de Planejamento a Longo Prazo baseado em teoria de grafos, desenvolvido nos **Capítulos** V e VI. **Soluções** para reservatórios de **água** foram propostos em [221] e [23] .

#### Seção 4 - CONCLUSÃO

O modelo apresentado é Suficientemente geral para representar problemas em **vários** sistemas descritos por redes. A particularidade mais restritiva de formulação está na fixação dos **parâmetros** dos nós, o que não pode ser feito em alguns sistemas, como se verá abaixo. A generalização para esses casos é imediata, bastando introduzir na definição de configuração um vetor de estado correspondente aos **nós**, definindo custos da transição de estados, conjuntos de estados **admissíveis** para os **nós**, etc. A formulação dos problemas deve ser modificada para incluir o planejamento do estado dos **nós**. Neste caso os **vínculos** podem assumir grande complexidade, podendo tornar-se **impossível** colocá-los no formato (11), o que não limita o modelo, pois basta mudar a definição de viabilidade.

A viabilidade de uma configuração é o ponto que mais depende do tipo de sistema considerado. Sistemas de transmissão de **potência** devem satisfazer ambas as leis de Kirschhoff em várias **situações** de emergência, como se verá no **próximo** capítulo. Se as injeções de potência (parâmetros dos nós) não forem fixas, esses **vínculos** deverão ser satisfeitos pelo fluxo resultante de um problema de **otimização** (despacho de carga), o que não se estudará neste trabalho. Esse caso é estudado em [14], restrito apenas pela primeira lei de Kirschhoff.

Sistemas de transporte [18] e sistemas de tráfego aéreo [10] são englobados pela formulação apresentada, sendo que os **vínculos** incluirão um problema de otimização (busca de fluxo **compatí**

## 11.4

vel). Por outro lado, somente a primeira lei de Kirschhoff deve ser satisfeita.

Sistemas de transporte de **gás** são semelhantes aos sistemas elétricos, admitindo modelos plenamente adaptados **à** **fórmulação** proposta [10] .

Sistemas de **reservatórios** de água interligados admitem modelos simples [20] , em que o estado de uma configuração se refere aos **nós** e não aos ramos. As condições de viabilidade podem, no entanto, ser complexas, suscitando problemas de decisão não **seriada**, **resolvidos** em [20] por meio de programação dinâmica não seriada [21] . A otimização a longo prazo desses sistemas **é** estudada em [22] e [23] .

A **formalização** apresentada neste capítulo será utilizada no Capítulo VI, onde se aplicará ao problema de expansão de capacidade em redes o **algoritmo** de decisão sequencial introduzido no Capítulo V . Os dois **próximos** capítulos tratarão da resolução completa do problema de **otimização** a curto prazo.

## CAPÍTULO III

### OTIMIZAÇÃO A CURTO PRAZO

Neste capítulo estuda-se a expansão de uma configuração  $c^0$  de uma rede básica, dados os **parâmetros** dos nós. Com esta finalidade, **constói-se** um grafo  $H$ , **cujos nós** são configurações, entre as **quais** encontram-se as sucessoras de  $c^0$ . O problema de busca das sucessoras fica portanto reduzido a um problema de busca de caminhos de custo **mínimo** através de  $H$ .

No entanto, a busca de todas as sucessoras de  $c^0$  não apresenta interesse **prático** em geral, uma vez que instalações adicionais **desnecessárias** levam geralmente a novas sucessoras **excessivamente** custosas. Define-se então o conceito de sucessora **eficiente**, e desenvolve-se um algoritmo capaz de encontrar um **número desejado** de tais sucessoras. Se o **número** de sucessoras desejado for 1, o algoritmo resolve o problema de **otimização** a curto prazo. Se não houver **limitação**, executa-se a expansão "**eficiente**" de  $c^0$ .

O **algoritmo** utilizado baseia-se na **idéia** de **algoritmos** de busca "**em profundidade**" (depth first methods), sugeridos em [2], baseados em um procedimento conhecido por "**Backtrack Programming**" [35], e semelhante ao algoritmo de Trémaux, descrito em [18]. Trata-se de um **algoritmo** em que se procura **minimizar** a **memória** de computador ocupada, com **prejuízo** da eficiência. A economia de memória é importante em planejamento, uma vez que a expansão deverá operar como uma subrotina de um programa de **otimização** a longo pra.

III.

zo.

A seção 1 ocupa-se do enunciado e da formulação de um algoritmo básico, cuja admissibilidade é demonstrada em seguida. A seção 2 especializa o algoritmo para o problema de expansão e comenta sua eficiência.



111.1

## Seção 1 - O PROBLEMA DE EXPANSÃO

- 1 Seguindo a notação do Capítulo II, considerem-se dados uma configuração inicial  $c^0 = (N, M, S^0)$  para uma rede básica  $(\bar{N}, \bar{M})$ , um vetor de parâmetros  $P \in R^n$  e um estágio  $t \in T$ .

Suponham-se conhecidos todos os elementos introduzidos no Capítulo II, necessários ao estudo de custos e viabilidade de configurações.

Estuda-se a expansão de  $c^0$  com respeito à topologia  $(N, M)$  e ao vetor de parâmetros  $P$  no estágio  $t$ , conforme 11.2.17. Observe-se que não se perde generalidade fixando a topologia em  $(N, M)$ , pois se se tivesse  $c^0 = (N^0, M^0, \hat{S}^0)$  com  $m_0 < m$ , bastaria definir  $s_p^0 = \bar{s}_{i_p}$  para  $p = m_0 + 1, \dots, m$ , segundo as convenções em 11.2.14.

## 2 O Grafo de expansão

O grafo de expansão é um grafo simples  $H$ , cujos nós são configurações, e que admite como centro o nó  $c^0$ .  $H$  fica totalmente definido por  $c^0$  e por seu operador  $A$  (Ver A.12).

- 3 Definição : O grafo de expansão  $H = (N_H, \Lambda)$  é definido recursivamente por:

(i)  $c^0 \in N_H$

(ii) Sejam  $c = (N, M, S) \in N_H$  e  $c' = (N, M, S')$

Então  $c' \in \Lambda(c)$  se e somente se

(a)  $c'$  é uma configuração

(b)  $(S, S')$  é uma transição de estado elementar (definida em (11.2.21)).

111.1

- 4 Observação :  $H$  é finito, pois o conjunto de estados admissíveis  $a$  é finito. No entanto, o grafo pode ter porte muito grande e admite circuitos em geral.
- 5 Qualquer sequência de transições de estado elementares a partir de  $C \in N_H$  corresponde a um caminho em  $H$ .
- 6 Custos : Cada ramo  $(C^1, C^2)$  de  $H$  corresponde a uma transição elementar de estado  $(S^1, S^2)$  na rede. Associa-se naturalmente a cada ramo de  $H$  o custo de transição de estado  $c_H(C^1, C^2) \triangleq c_t(C^1, C^2, t)$  definido em 11.2.25.
- 7 Dadas duas configurações  $C^1, C^2 \in N_H$ , o custo de transição  $c_t(C^1, C^2, t)$  será dado pelo custo de um caminho de custo mínimo de  $C^1$  para  $C^2$  através de  $H$ , se houver algum tal caminho;  $c_t(C^1, C^2, t) = +\infty$  no caso contrário. Este fato é consequência imediata da definição de  $c_t$  em 11.2.22 e da observação (5). Da definição de  $c_t$  conclui-se ainda que não há circuitos de custo negativo em  $H$  e que sempre é possível encontrar um caminho de custo mínimo entre  $C^0$  e qualquer  $C \in N_H$ .

### Expansão de $C^0$

- 8 O conjunto de sucessores de  $C^0$ , definido em 11.2.17 é dado por:

$$\Gamma(C^0, (N, M), P, t) = \{C \in N_H \mid C \text{ é viável com respeito a } P \text{ e } S \in \delta(S^0, t)\}$$

De fato, qualquer sucessora de  $C^0$  pode ser construída por uma sequência de transições de estado elementares a partir de  $C^0$  e portanto  $\Gamma(C^0, (N, M), P, t) \subset N_H$ . A igualdade acima segue trivialmente deste fato e da definição de sucessora.

Como já se comentou, a descrição completa de  $\Gamma(C^0, (N, M), P, t)$  não tem interesse prático, pois algumas sucessoras podem incluir equipamento desnecessário. Define-se a seguir o conceito de sucessora ineficiente: a definição é geral e a verificação da ineficiência de uma sucessora depende do sistema em consideração, sendo imediata no caso de sistemas de potência (Ver VI.2 . 28 )

- 9 Definição : Uma configuração  $C \in \Gamma(C^0, (N, M), P, t)$  é ineficiente se existir  $C' \in \Gamma(C^0, (N, M), P, t)$  tal que  $C'$  pertence a um caminho de custo mínimo de  $C^0$  a  $C$ , com  $c_t(C^0, C, t) > c_t(C^0, C', t)$ .  
 $C$  é eficiente se não for ineficiente.

- 10 O conjunto de sucessores eficientes de  $C^0$  será denotado por  $\hat{\Gamma}(C^0, t)$ . A cada configuração sucessora de  $C^0$  associa-se naturalmente o custo  $c(C^0, C, P, t)$  definido em 11.2.25.

Para que a definição de eficiência seja consistente, é necessário que não se possa reduzir o custo total de um sistema por meio da construção de instalações desnecessárias, ou seja, economias no custo de operação não devem justificar a construção de instalações desnecessárias quanto à viabilidade. Esta hipótese é formalizada abaixo:

- 11 Hipótese : Sejam  $C$  e  $C'$  configurações como em (9), sendo  $C$  ineficiente. Então

$$c(C^0, C, P, t) > c(C^0, C', P, t)$$

Se esta hipótese não for satisfeita, a definição de eficiência deve ser modificada, de modo a incluir a condição (11), o que não invalida o tratamento a seguir. Algumas modificações deverão ser feitas no algoritmo de expansão, em especial quanto à simpli-

## III.1

ficação (15) .

Com os elementos acima, pode-se enunciar o problema de expansão generalizado, cuja importância é evidente e será explorada no estudo de otimização a longo prazo.

## 12 Problema de Expansão Generalizada

PEG: Dada uma configuração  $C^0$ , um estágio  $t$  e um vetor de parâmetros correspondente a  $C^0$  e  $t$ , dado  $k > 0$ , inteiro, encontrar um conjunto

$$\hat{\Gamma}_k(C^0, t) \subset \hat{\Gamma}(C^0, t) \text{ tal que } |\hat{\Gamma}_k(C^0, t)| = \min\{k, |\hat{\Gamma}(C^0, t)|\} \text{ e}$$

$$c(C^0, C, P, t) \leq c(C^0, C', P, t) \quad \forall C \in \hat{\Gamma}_k(C^0, t), \quad \forall C' \in \hat{\Gamma}(C^0, t) - \hat{\Gamma}_k(C^0, t)$$

Encontrar também os custos associados aos elementos de  $\hat{\Gamma}_k(C^0, t)$

Em outras palavras, procuram-se  $k$  sucessores de mais baixo custo entre todas as sucessoras eficientes.

13 Observação : Se  $k=1$ , PEG corresponde ao problema de otimização a curto prazo 11.2.29 . Se  $k=\infty$ , PEG corresponde ao problema de expansão, com eliminação das sucessoras ineficientes.

O problema PEG é equivalente ao problema de construir uma arborescência mínima em  $H$  com centro em  $C^0$ , incluindo<sup>1</sup> como nós os elementos de algum conjunto  $\hat{\Gamma}_k(C^0, t)$  definido em (11). O problema é complicado pela presença de custos negativos e pelos custos de operação, mas pode, em princípio, ser resolvido por Programação Dinâmica, por algoritmos de busca em grafos ou pelo algoritmo desenvolvido no Capítulo V .

## III.1

A presença de custos negativos é pouco comum na prática, e ocorre quando se considera a revenda de instalações, sem substituí-las por outras. Esse caso será desprezado no que segue, adotando-se a seguinte hipótese:

- 14 Hipótese : Sejam  $c^1 \neq c^2$  duas configurações quaisquer de mesma topologia. Então,  $\forall t \in T$

$$c_t(c^1, c^2, t) > 0$$

A hipótese justifica a seguinte redução do grafo  $H$  :

- 15 Simplificação de  $H$  : Na definição de  $H$  (3), impõe-se a seguinte condição:

Se  $C$  é sucessora de  $c^0$ , então  $\Lambda(C) = \emptyset$

De fato, a simplificação acima não modifica a solução de PEG, uma vez que nenhum caminho de custo mínimo entre  $c^0$  e uma sucessora eficiente  $C$  pode passar por outra sucessora  $C'$ : isso entraria em contradição com a definição de eficiência, pois  $c_t(C', C, t) > 0$  devido a (14).

A hipótese de positividade (14) permite a utilização de métodos mais eficientes para a resolução de PEG, como o algoritmo de Dijkstra [3]. No entanto, tanto este método como Programação Dinâmica ocupam muita memória de computador. O problema torna-se crucial quando a descrição completa de uma configuração é muito complexa, como ocorre no estudo de sistemas de transmissão de potência, em que a matriz de capacidades de cada configuração deverá residir na memória (Ver próximo capítulo).

## III.1

Desenvolve-se a seguir um algoritmo de busca de caminhos de custo mínimo entre um nó inicial e todos os nós de um alvo  $T$  em um grafo finito e sem ciclos de custo negativo. Uma adaptação do algoritmo fornecerá um método para a resolução do Problema de Expansão Generalizado, o que será objeto da seção seguinte.

- 16 Considere-se um grafo  $H = (N_H, \Lambda)$ . Sejam  $c^0 \in N_H$ ,  $T \subset N_H$  e seja  $p > 0$  um inteiro. O seguinte algoritmo, sugerido em [2], pesquisa caminhos em  $H$  até o comprimento máximo  $p$  (o comprimento de um caminho é dado pelo número de ramos que o formam). O algoritmo mantém duas listas de nós, acompanhados de custos e indicadores, chamadas respectivamente lista Aberto (cujos nós são abertos) e lista Fechado (composta de nós fechados). A cada nó  $c$  será também associado um número  $n(c)$  correspondente à sua profundidade no grafo.

Uma terceira lista, a Lista de Sucessores, com a mesma estrutura das anteriores, será manipulada pelo algoritmo.

- 17 Algoritmo : Inicialmente, Aberto, Fechado e a Lista de Sucessores estão vazias .

Passo 0

Coloque  $c^0$  em Aberto, associando-lhe  $\hat{g}(c^0)=0$ ,  $n(c^0)=0$

Passo 1

Se houver algum nó em Aberto, retire dessa lista o primeiro nó  $\bar{c}$  e passe-o para Fechado.

Vá para o passo 2 .

Senão, termine.

111.1

Passo 2

Se  $\bar{c} \in T$ , guarde  $\bar{c}$  na lista de Sucessores. Se  $\bar{c}$  já se encontra nessa lista, associe a  $\bar{c}$  o menor entre os custos encontrados, atualizando o apontador.

Vá para o passo 1.

Senão, se  $n(\bar{c}) = p$ , vá para o passo 1.

Senão, continue.

Passo 3

Expanda  $\bar{c}$ , encontrando  $\Lambda(\bar{c})$ .

Para cada  $c \in \Lambda(\bar{c})$ , calcule

$$\hat{g}(c) = \hat{g}(\bar{c}) + c_H(\bar{c}, c)$$

$$n(c) = n(\bar{c}) + 1$$

Retire de  $\Lambda(\bar{c})$  todos os elementos já listados em Aberto ou Fechado com profundidade e custo não superiores aos valores calculados agora.

Introduza no início de Aberto os nós remanescentes em  $\Lambda(\bar{c})$ , associando-lhes apontadores  $\bar{c}$ .

Para estudar a operação do algoritmo (17), deve-se inicialmente observar o seguinte:

- 18 Lema : Sejam  $c, \bar{c}$  dois elementos presentes em Aberto em um dado instante. Se  $n(c) < n(\bar{c})$ , então  $c$  está colocado em Aberto em uma posição posterior à de  $\bar{c}$ .

Demonstração : O passo 3 sempre coloca sucessores no início da lista. Seja  $\bar{c} \in \Lambda(\bar{c}')$ . Todos os elementos de  $\Lambda(\bar{c}')$  tem nível de profundidade  $n(\bar{c})$ , por construção, e são colocados no início da lista. Enquanto existirem na lista Aberto elementos de

$\Lambda(\bar{C}')$ , só poderão ser gerados nós com nível maior do que  $n(\bar{C})$ , pois o passo 1 sempre escolhe o primeiro elemento de Aberto e o passo 3 incrementa o nível de  $n(C)$ ,

Consequentemente, só podem estar em posições anteriores à de  $\bar{C}$  em Aberto, elementos  $C$  com  $n(C) \geq n(\bar{C})$ , o que completa a demonstração.

19 Como consequência de (18), a lista Aberto pode ser particionada em um conjunto de  $p$  listas, denotadas daqui por diante por  $O^i$ ,  $i=1,2,\dots,p$ , denominadas listas de opções, cada uma correspondente a um nível de profundidade. A lista Aberto tem uma estrutura de pilha, em que se superpõem  $O^k, O^{k-1}, \dots, O^1$ ,  $k \leq p$ , com  $O^k$  nas primeiras posições.

20 Os elementos de cada lista  $O^i$  são sucessores de um mesmo nó, retirado de  $O^{i-1}$ ; quando uma lista  $O^i$  é gerada, estão vazias as listas  $O^{i+1}, \dots, O^p$ .

22 Teorema : O algoritmo termina em tempo finito, mesmo que não se façam as eliminações do passo 3.

### Demonstração

A cada nó que entra em Aberto está associado um caminho de comprimento não superior a  $p$ , devido ao passo 2.

Mostra-se que nunca entra em Aberto um nó  $C^n$  associado a um caminho já encontrado:

De fato, suponha-se que  $C^n$  entrou duas vezes em Aberto, associado, respectivamente, aos caminhos  $G = (C^0, C^1, \dots, C^{n-1}, C^n)$ ,  
 $\hat{G} = (C^0, \hat{C}^1, \dots, \hat{C}^{n-1}, C^n)$ .



## III.1

Após  $c^n$  entrar em Aberto, uma nova entrada de  $c^n$  com o mesmo nível de profundidade só pode dar-se após esgotarem-se os elementos de  $\Lambda(c^{n-1})$ , colocados no início da lista. Como  $c^{n-1}$  está em Fechado, na nova ocorrência  $c^n \in \Lambda(\hat{c}^{n-1})$ , com  $\hat{c}^{n-1}$  retirada de Aberto pelo passo 1.

Se  $G = G$ , então  $c^{n-1}$  entrou duas vezes em Aberto associado ao mesmo caminho.

Por recursão,  $c^1$  entrou em Aberto duas vezes, o que é absurdo, pois  $c^0$  só é expandido uma vez.

Como o número de caminhos partindo de  $c^0$ , com comprimento menor ou igual a  $p$ , é finito, o algoritmo pára em tempo finito.

- 23 **Teorema** : Ao terminar o algoritmo, encontram-se na lista de Sucessores todos os nós  $C$  de  $T$ , tais que existe um caminho ótimo entre  $c^0$  e  $C$  com comprimento menor ou igual a  $p$ . Os custos associados a esses elementos da lista correspondem ao custo de um caminho mínimo a partir de  $c^0$ .

Demonstração :

Seja  $C \in T$  acessível a partir de  $c^0$  por um caminho de custo mínimo.

$$G = (c^0, c^1, \dots, c^q) \quad q \leq p, \quad c^q = C, \quad c^i \notin T, \quad i=1, 2, \dots, q-1$$

- 24 Mostra-se que  $c^1, c^2, \dots, c^q$  estão na lista Fechado, com

$$\hat{g}(c^i) = c_H(c^0, c^1, \dots, c^i) \quad i=1, 2, \dots, q$$

Tem-se, por indução:

$c^1$  está em Fechado com  $\hat{g}(c^1) = c_H(c^0, c^1)$ , pois foi gerado na primeira iteração, sendo fechado posteriormente (uma vez que Aberto está vazia)

## III.1

Suponha-se que (24) é válida para  $c^1, c^2, \dots, c^k$ ,  $1 \leq k < q$

Como  $c^k \notin T$ , gera-se  $c^{k+1}$  na iteração em que  $c^k$  é fechado, com

$$\hat{g}(c^{k+1}) = \hat{g}(c^k) + c_H(c^k, c^{k+1}) = c_H(c^0, c^1, \dots, c^{k+1})$$

Como esse custo é mínimo por hipótese, o passo 3 conserva  $c^{k+1}$  em alguma lista com esse custo.

Pelo teorema (22),  $c^{k+1}$  será fechado em alguma iteração, o que demonstra (24).

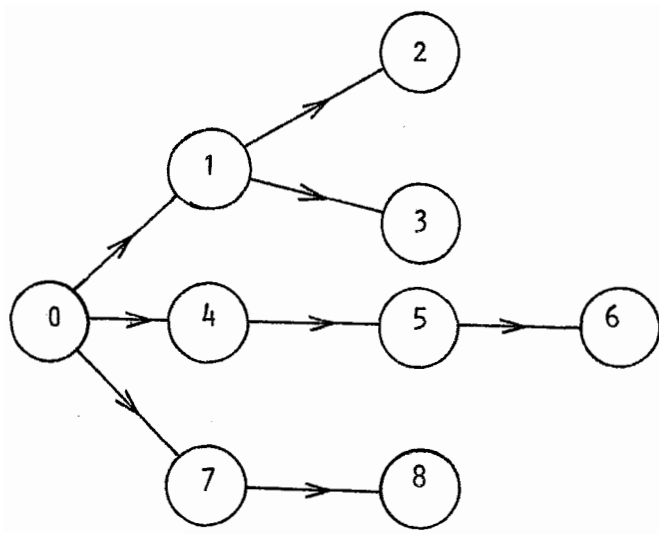
Completa-se a demonstração, observando que  $c^q$  entra na lista de Sucessores ao ser fechado (se já não estiver na lista associada ao mesmo custo).

A operação do algoritmo consiste, portanto, no seguinte: inicia-se percorrendo um caminho em  $H$ , gerando as listas  $c^0, c^1, c^2, \dots$ , até encontrar um nó de  $T$ , ou até ser atingido o limite de profundidade  $p$ . Retorna-se então pelo caminho encontrado, gerando outro caminho com o maior número possível de nós em comum com o anterior.

Como exemplo, a árvore da figura III.1 seria percorrida na ordem indicada em seus nós. (Veja na próxima página).

As observações acima serão úteis na próxima seção, ao adaptar-se o algoritmo (17) à resolução de PEG.

III.1



**Figura III.1 :** O algoritmo somente abandona um nó ao serem gerados todos os caminhos emergentes desse nó.

## Seção 2 - RESOLUÇÃO DO PROBLEMA DE EXPANSÃO GENERALIZADO

Procura-se na resolução de PEG economizar ao máximo a memória ocupada pelo algoritmo. A descrição completa de uma configuração pode ser muito complexa e a verificação de suas condições de viabilidade podem exigir conjuntos de parâmetros ou matrizes de grande porte. A observação (20) fornece uma boa ferramenta para a minimização da memória, quando os dados para uma configuração podem ser obtidos a partir dos dados de sua genitora; pode-se, nesse caso, guardar somente um conjunto de dados por nível de profundidade, identificando cada elemento de Aberto apenas pelo seu estado, ou pela transição de estado que a gera a partir da genitora. Em problemas de planejamento, transições elementares de estado costumam ter uma expressão muito simples. Em sistemas de transmissão de potência, por exemplo, uma transição elementar é especificada pelo índice do ramo a ser reforçado e pelo tipo de linha a ser construída, como se estudará no próximo capítulo.

- 25 Lista Fechado : A lista Fechado consiste de configurações, acompanhadas de seu custo e do nível de profundidade. Basta guardar o mínimo de informação sobre o estado de cada configuração suficiente para fazer comparações com configurações geradas. É importante observar que a lista Fechado pode não existir, se não houver memória suficiente, perdendo-se em eficiência: de fato, nada muda na demonstração dos teoremas (22) e (23) se se limitar o comprimento de Fechado. A lista Fechado é desnecessária ao terminar o algoritmo, pois não se deseja recuperar caminhos em  $H$  : deseja-se somente determinar as configurações sucessoras de  $C^0$  (Ver 12) . Bons

## III.2

resultados foram obtidos limitando Fechado a um comprimento pré-de terminado, usando um método não determinístico para substituir elementos de Fechado após a lista estar completa.

- 26 Lista de Sucessoras (L): Nessa lista devem-se guardar descrições completas de configurações, principalmente quando a lista Fechado é incompleta, sendo conveniente limitá-la ao máximo. No problema de Expansão Generalizada desejam-se  $k$  sucessoras eficientes, no máximo. No algoritmo proposto a seguir, limita-se o comprimento dessa lista a  $k$  elementos, introduzindo-se um novo elemento somente se o seu custo for inferior ao maior custo presente na lista.

- 27 Isto permite que as listas de opções sejam truncadas: se a lista de sucessoras estiver completa e se  $\bar{c}$  for o maior custo presente na lista, podem-se retirar da lista Aberto todas as configurações  $\bar{c}$  com  $\hat{g}(\bar{c}) \geq \bar{c}$ ; de fato, se algum caminho  $G$  de custo inferior a  $\bar{c}$  entre  $c^0$  e  $T$  passa por  $\bar{c}$ , a parte (b) da demonstração do teorema (18) mostra que em alguma iteração  $\bar{c}$  entrará em Aberto com

$$\hat{g}(\bar{c}) \leq c_H(G) < \bar{c}$$

Isso justifica a retirada de  $\bar{c}$  da lista Aberto, uma vez que  $\bar{c}$  voltará a ser gerada se estiver em um caminho de custo inferior a  $\bar{c}$ .

A introdução de elementos na lista de Sucessores deve também eliminar sucessoras ineficientes, segundo (9).

- 28 Listas de Opções: Ao invés de retirar um nó  $c^n$  da lista de Opções  $n^n$  em que se encontra (e portanto, da lista Aberto), o pas-

## III.2

so 1 conservará a lista, incrementando um indicador  $o_n$  associado ao nível de profundidade de  $c^n$ . Os elementos de  $o^n$  de índice superior a  $o_n$  serão abertos, e o elemento  $o_{o_n}^n$  tem como sucessores os nós em  $o^{n+1}$ . Segundo as observações feitas acima, somente a configuração na posição  $o_{o_n}^n$  deverá ter sua descrição completa na memória.

- 29 Custos de operação : Os custos de operação devem ser adicionados ao custo de um caminho terminado por uma sucessora. Isso é feito acrescentando esse custo ao custo do último ramo do caminho.

O algoritmo a seguir, no caso em que só se busca uma sucessora (resolvendo portanto PCP) e a lista Fechado é completa, corresponde ao algoritmo de Programação Dinâmica Truncada, exposto em [14].

- 30 Algoritmo : São dados os elementos necessários ao enunciado de PEG(12) e é dada uma profundidade máxima  $p \geq 1$ .

Passo 0

Se  $c^0$  for viável, faça  $c^0 = c_o(c^0, p, t)$  e termine com a lista de sucessoras  $L = \{(c^0, c^0)\}$

Senão, faça  $n=0$ ,  $o_n=0$  para  $n=1, 2, \dots, p$ ,  $c^0=0$ ,  $\bar{c}=+\infty$ ,  $L=\emptyset$ .

Vã para o passo 3.

Passo 1

Faça  $o_n = o_n + 1$

## III.2

Se  $o_n \leq l_n$  obtenha, a partir de  $C^{n-1}$ , a descrição completa de  $C^n$ , onde  $C^n$  é o elemento do índice  $o_n$  da lista  $O^n$ .

Introduza  $C^n$  na lista Fechado, associando-lhe  $\hat{g}(C^n)$  e o nível  $n$ .

Vá para o passo 2.

Senão, se  $n=1$ , termine

senão, faça  $o_n=0$ ,  $O^n=\emptyset$ ,  $n=n-1$  e vá para o passo 1

Passo 2

Se  $C^n$  for sucessora de  $C^0$ , faça  $c = \hat{g}(C^n) + c_0(C^n, p, t)$ ,  $C = C^n$  e guarde  $(C, c)$  na lista de Sucessoras, conforme o procedimento (32) ou (33). Vá para o passo 1.

Senão, se  $n=p$ , vá para o passo 1.

senão, continue.

Passo 3

Determine  $\Lambda(C^n)$ , segundo a definição (3).

Para cada  $C \in \Lambda(C^n)$ , calcule

$$\hat{g}(C) = \hat{g}(C^n) + c_t(C^n, C, t)$$

Introduza em  $O^{n+1}$  os elementos de  $\Lambda(C^n)$ , acompanhados dos custos calculados

Retire de  $O^{n+1}$  os elementos  $C$  tais que  $\hat{g}(C) \geq \bar{c}$

Retire de  $O^{n+1}$  os elementos  $C$  já listados, associados a níveis e custos menores ou iguais a  $n$  e  $\hat{g}(C)$ , respectivamente.

Faça  $l_{n+1} = |O^{n+1}|$ ,  $n=n+1$  e vá para o passo 1.

## 111.2

31 Lista de Sucessoras :

Trata-se de uma lista  $L$  , cujos elementos são pares  $(c^i, c^i)$  , onde

$$(c^i, c^i) , \quad i=1,2,\dots,|L| , \quad |L| \leq k$$

correspondem em cada iteração a configurações sucessoras de  $c^0$  e custos  $c^i \leq \bar{c}$  .

32 Introdução de  $(C, c)$  na lista  $L$  , quando  $k=1$ 

Se  $c < \bar{c}$  , faça  $(c^1, c^1) = (C, c)$  ,  $\bar{c}=c$  .

33 Introdução de  $(C, c)$  na lista  $L$  quando  $k>1$ 

(a) Se  $c \geq \bar{c}$  ou se algum elemento de  $L$  elimina  $(C, c)$  pela regra (34), retorne ao algoritmo

Senão, continue.

(b) Retire de  $L$  todos os elementos eliminados por  $(C, c)$  pela regra (34)

(c) Introduza  $(C, c)$  na lista  $L$  .

Se  $|L| = k+1$  , retire de  $L$  um elemento de custo máximo.

Faça  $\bar{c} = \max\{c^i , i=1,2,\dots,k \}$

34 Regra de eliminação de sucessoras ineficientes

$(C, c)$  elimina  $(C', c')$  se

(i)  $c_t(C^0, C', t) \geq c_t(C^0, C, t) + c_t(C, C', t)$  , ou se

(ii)  $C = C'$  ,  $c' \geq c$

Ou seja, elimina-se uma configuração quando sua ineficiência é detectada.



## III.2

35 Admissibilidade do algoritmo (30):

Verifica-se facilmente que o algoritmo (30) corresponde a uma implementação do algoritmo (17), ao qual se acrescentam as eliminações por custo comentadas em (27).

O passo 0 de (30) inicializa o algoritmo, sem colocar  $c^0$  em Fechado, o que é desnecessário.

No passo 1, o algoritmo (30) cicla até encontrar um nó aberto, ou termina se Aberto estiver vazia. De fato, em qualquer iteração, a lista Aberto é composta pelos elementos das listas  $O^i$  de índices superiores a  $o_i$ ,  $i=1,2,\dots,p$ . O passo 1 inicia com o último nível de profundidade pesquisado (somente o passo 3 incrementa  $n$  ao compor uma nova lista  $O^n$ ) e nesse instante  $O^{n+1} = O^{n+2} = \dots = O^p = \emptyset$ , se  $n < p$ .

O nível é decrescido pelo passo 1 até encontrar-se uma configuração em Aberto ou termina o algoritmo com  $O^i = \emptyset$ ,  $i=2,3,\dots,p$ ;  $o_1 = l_1$ .

O restante dos algoritmos é idêntico, observando-se que  $n$  em (30) acompanha sempre o valor de maior nível de profundidade com lista de opções não vazia, e que eliminam-se configurações de custo excessivo, o que é comentado a seguir.

36 O algoritmo termina, devido a (22). Suponha-se que nesse instante, o valor de  $\bar{c}$  é  $\hat{c}$ , e a lista de sucessoras é  $\hat{L}$ ,  $|\hat{L}| = k$ .

37 Lema : Se  $\bar{c}$  é uma sucessora eficiente acessível a partir de  $c^0$  por um caminho de custo mínimo de comprimento menor que  $p+1$ , e se

## III.2

$c(c^0, \bar{c}, P, t) < c$  , então :  $(\bar{c}, c(c^0, \bar{c}, P, t)) \in \hat{L}$  .

Demonstração:

Seja  $T$  o conjunto de todas as sucessoras eficientes  $c$  de  $c^0$ , tais que  $c(c^0, c, P, t) < c$  . A regra de eliminação (34) nunca elimina uma sucessora eficiente e (34ii) simplesmente duplica uma comparação existente no algoritmo (17). A comparação com  $\bar{c}$  no passo 3 de (30) nunca elimina um caminho mínimo entre  $c^0$  e  $T$ , como se mostrou em (27) (observe-se que em qualquer iteração  $\bar{c} \geq \hat{c}$ , pois  $\bar{c}$  nunca cresce) .

Potanto, a lista de sucessoras gerada equivale à lista construída pelo algoritmo (17), e o lema corresponde ao teorema (23), o que completa a demonstração.

38 Lema ; Se  $(c, c) \in \hat{L}$  , e se existe um caminho Ótimo de comprimento não superior a  $p$  entre  $c^0$  e qualquer sucessora eficiente , então  $c$  é uma sucessora eficiente de  $c^0$  e

$$c(c^0, c, P, t) = c$$

Demonstração:

(a)  $c$  é sucessora eficiente:

$c$  é sucessora por construção, devido ao passo 2

Se  $c$  não fosse eficiente, existiria  $\hat{c}$  eficiente, tal que

$$c(c^0, \hat{c}, P, t) + c(\hat{c}, c, P, t) = c(c^0, c, P, t)$$

Pelo lema (37),  $(\hat{c}, c(c^0, \hat{c}, P, t))$  estaria em  $\hat{L}$  , pois

$$c(c^0, \hat{c}, P, t) < c \leq \hat{c} \text{ e } c \text{ tem profundidade inferior a } p$$

em um caminho Ótimo

Isso contradiz a hipótese de  $c$  estar na lista de sucessoras,

## III.2

uma vez que  $C$  seria eliminada pela regra (34) ao ser fechada a Última entre as configurações  $C$  e  $\hat{C}$ .

$$(b) \ c(C^0, C, P, t) = c$$

Se se tivesse  $c(C^0, C, P, t) < c < \hat{c}$ , então  $C$  estaria na lista  $\hat{L}$  com custo  $c(C^0, C, P, t)$ , devido ao lema (37). A regra (34) impede a existência em  $\hat{L}$  de  $C$  associada a dois custos distintos. Portanto,  $c(C^0, C, P, t) \geq c$ . Evidentemente,  $c(C^0, C, P, t) \leq c$ , o que completa a demonstração.

Não é possível garantir que  $\hat{k}=k$ , pois várias eliminações em  $L$  podem ser feitas em estágios finais do algoritmo. A união dos dois lemas acima fornece o seguinte resultado:

39 **Teorema** : Com as convenções (37), se qualquer sucessora eficiente for acessível a partir de  $C^0$  por um caminho ótimo de comprimento menor ou igual a  $p$ , então  $\hat{L}$  tem o formato  $\hat{\Gamma}_k(C^0, t)$ , ou seja,  $L$  resolve o Problema de Expansão Generalizada (12) para  $k=\hat{k}$ .

O teorema (39) informa sobre os resultados que se pode esperar do algoritmo (30). A limitação de profundidade  $p$  visa apenas à economia de tempo de processamento e pode ser bastante relaxada se as listas de opções forem bem construídas, como se comenta abaixo. É possível também iniciar com  $p$  grande, reduzindo-o assim que sucessoras são encontradas pelo algoritmo.

Um aspecto essencial ao desempenho do algoritmo está ligado à construção das listas  $O^i$ .

## III.2

40 Observações: listas de opções

No tratamento acima, as listas  $0^i$  descrevem todas as **transições** de estado elementares a partir de  $c^{i-1}$ . As **observações** a seguir **são** essenciais para a eficiência do algoritmo.

(i) **Ordenação** das listas

Os elementos das listas devem ser ordenados segundo algum critério de eficiência. Uma primeira abordagem consiste em ordená-las por ordem de custo das transições correspondentes. Outros **critérios** podem apontar as transições mais "**promissoras**", o que depende do sistema e **será** usado em sistemas elétricos em IV, . Essa **ordenação** leva à obtenção de caminhos de custo pequeno no início da aplicação do **algoritmo**, aumentando as eliminações nos passos 2 e 3.

## (ii) Truncagem das listas

Guardam-se nas listas somente transições que possam levar a sucessoras, eliminando-se opções irrelevantes para a **viabilização** de  $c^0$ , dependendo do sistema em consideração. Este caso ocorre em problemas de transporte, quando necessariamente devem ser reforçados ramos de um cut-set (Ver [14]).

Esse tipo de truncagem não se aplica a sistemas de transmissão de potência, quando se considera a segunda lei de Kirchhoff, e usa-se o procedimento seguinte:

(iii) Procedimento **heurístico** de truncagem das listas

Quando se dispõe de um critério **capaz** de apontar as mais "**promissoras**" entre as opções em  $0^i$ , limita-se heuristicamente o comprimento das listas, ou a um valor previamente estipulado, ou por algum outro **critério** de eficiência. Um tal critério para sistemas e

## III.2

lêtricos será desenvolvido na seção IV. 4 . Neste caso, o teorema (39) não pode ser demonstrado e o resultado obtido será tanto melhor quanto mais eficiente for o critério de truncagem.

41 observação : Lista de Sucessoras

Embora o número de sucessoras desejado ao aplicar-se o algoritmo seja  $k$ , pode-se apenas garantir a obtenção de um número  $\bar{k} \leq k$  de sucessoras. Isto ocorre porque ao final do algoritmo, vários elementos da lista podem ser eliminados simultaneamente pela regra (34). Conclui-se que, se a ordenação das listas de opções for bem feita, dificilmente se obterá  $\bar{k} < |\hat{r}_k(c^0, t)|$ . Somente é possível garantir  $\bar{k} = k$  se o algoritmo utilizar  $k = \infty$ , truncando-se a lista no comprimento desejado ao terminar. Isso é indesejável em geral, porque se perde a truncagem das listas de opções no passo 3, além de não se poder prever a quantidade de memória ocupada pela lista.

Uma versão do algoritmo (30) para o caso de sistemas de transmissão de energia elétrica é apresentada no capítulo seguinte e resultados obtidos com sua aplicação constam do Capítulo VII .

## CAPÍTULO IV

### MODELAGEM E EXPANSÃO DE SISTEMAS DE TRANSMISSÃO DE POTÊNCIA

Neste capítulo apresenta-se uma adaptação de sistemas de transmissão de potência ao modelo de redes introduzido no **capítulo II**, seguindo-se o estudo da aplicação a esses sistemas do **algoritmo** de expansão apresentado no capítulo **III**.

O ponto central da modelagem reside no modelo de **desempenho** do sistema, que **definirá** as condições de viabilidade de uma configuração.

O estudo do desempenho de sistemas de transmissão envolve grande complexidade, tratando-se de um campo de pesquisas em evolução constante, longe ainda de uma cristalização de conceitos e métodos. Nos estudos usuais de operações de sistemas de potência estuda-se, através de modelos rigorosos, a confiabilidade [26] ou o conceito moderno de controle de segurança [27] do sistema.

Esses estudos rigorosos, envolvendo testes de carga - precisos e estabilidade de máquinas interligadas, consomem um tempo de computação proibitivo para o emprego em planejamento a longo prazo. Por esta razão usam-se modelos simplificados para o fluxo de carga e para condições de estabilidade, expostos na seção 2. Não se pretende que os resultados de uma **otimização** a longo prazo sejam **implementadas** imediatamente: eles devem servir de guia ao estudo aprofundado da **implementação** usando modelos precisos, como se comenta - em [28].

O modelo de fluxo de carga será o modelo linear, exposto em [30], usado também em [9], [10], [11], [15]. Um modelo ainda mais simplificado encontra-se em [14], baseado **somente** na primeira lei de Kirschhoff, sem levar em conta a estabilidade do sistema.

## IV.

Este modelo no entanto, não fixa os **parâmetros** dos **nós** (injeções de potência), sendo o despacho de carga Ótimo um resultado fornecido - pelo modelo.

Um estudo de precisão do modelo linear [29] indica que os resultados são bastante **próximos** dos valores verdadeiros, **não se** , incorrendo portanto em erro excessivo.

A partir do modelo linearizado da rede de transmissão pode-se fazer um estudo de sensibilidade, **cujos** resultados serão - ferramentas eficientes para a construção de rotinas rápidas para a **análise** de contingências. Também a partir da análise de sensibilidade, desenvolve-se um critério para a ordenação e eliminação de elementos das listas de **opções** usadas pelo **algoritmo** de **otimização** a curto prazo.

Com esses elementos, particulariza-se na seção 3 o **algoritmo** de expansão , obtendo-se uma rotina de otimização a curto - prazo bastante eficiente, cuja implementação será comentada no **capítulo VII**.

## IV.1

Seção 1 - O Modelo do Sistema

Um sistema de geração e transmissão de potência é modelado por uma rede finita, cujos **nós** estão associados a pontos de carga ou geração de potência e cujos ramos estão associados a caminhos **permissíveis** para a construção de linhas de transmissão. Existe um **número** pequeno de tipos de linhas **permissíveis**, de acordo com sua tensão nominal e sua bitola (cinco tipos para o sistema do Sudeste brasileiro) e deve-se planejar a **sequência** Ótima de construção de linhas.

Passa-se agora à identificação dos elementos definidos no capítulo anterior, relativas ao sistema de transmissão.

- 1 Rede básica 11.1.2: trata-se da rede que inclui como **nós** todos os pontos de geração e carga **permissíveis** em todo o planejamento, e como ramos, todos os caminhos permissíveis (trechos) para a construção de linhas. Podem mesmo ser **incluídos nós** e ramos que nunca serão usados para **implementação**. Um exemplo encontra-se na figura IV.1, que representa a rede básica para o sistema Sudeste. A numeração dos **nós** partirá de zero, ou seja,  $\bar{N} = (\bar{n}_0, \bar{n}_1, \dots, \bar{n}_n)$ .
- 2 Estados permissíveis (11.1.4) : conhecidas as combinações de tipos de linhas cuja construção é permitida em cada trecho, constrói-se uma tabela que associa a cada combinação permitida em um ramo, um **número** inteiro. Uma tabela de transição de estados fornece, para cada ramo, o novo estado obtido pela adição de cada tipo de linha ao ramo no estado anterior. Um exemplo completo é descrito no capítulo VII.
- 3 Configuração (11.1.5): um exemplo de configuração para a rede básica da figura IV.1 está representado na figura V11.2, em que o estado de cada ramo é indicado graficamente. Outros exemplos estão no capítulo VII.



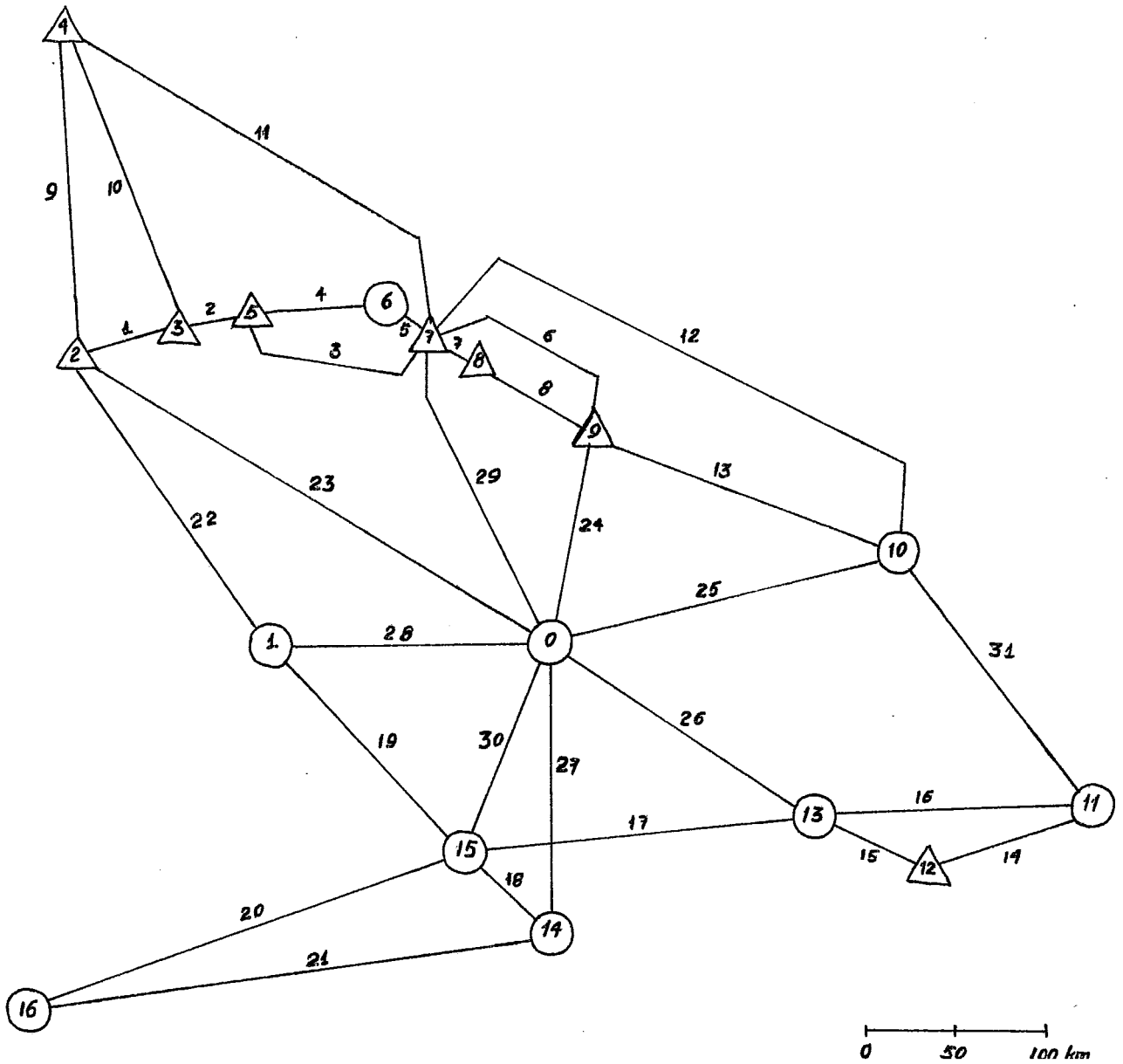


Fig. IV. 1 - Configuração Básica para o Sistema FURNAS.

## IV.1

- 4 Parâmetros dos nós (II.1.9):  $P_j$ ,  $j = 0, \dots, n$ , dado pela potência ativa injetada no nó  $n_j$  de uma configuração.  $P_j$  será positivo ou negativo, segundo haja geração ou carga no nó  $n_j$ .

Modelo de desempenho: as condições de viabilidade são desenvolvidas a seguir, resultando no critério de segurança (20). O modelo de fluxo de carga linear será apenas exposto sem justificativa, uma vez que se trata de um estudo bem conhecido [10,29]. A notação seguirá [10], onde encontram-se também as demonstrações da validade dos resultados.

- 5 As seguintes hipóteses são feitas com respeito ao sistema de transmissão de potência:
- (i) As tensões são constantes em cada nó, iguais ao seu valor nominal.
  - (ii) Não há perdas de potência nas linhas (hipótese usada apenas para a obtenção do modelo de desempenho, e não para o modelo de custos).
  - (iii) As defasagens nos ramos são suficientemente pequenas para se aproximar funções trigonométricas por seus arcos.

Seja  $C = (N, M, S)$  uma configuração qualquer para o sistema de potência. Suponhamos que  $|N| = n+1$ ,  $|M| = m$ ,  $N = (n_0, n_1, \dots, n_n)$ . Suponhamos ainda que os ramos são da forma  $r_i = (n_r, n_p)$ , com  $p > k$

O sistema descrito por  $C$  será modelado por uma rede dotada de:

- 6 fluxo:  $f \in \mathbb{R}^m$ , onde  $f_i$  representa o fluxo de potência elétrica através do ramo  $i$ ,  $i = 1, 2, \dots, m$
- 7 potencial:  $\theta \in \mathbb{R}^n$ , onde  $\theta_j$  representa a fase da tensão elétrica no nó  $n_j$ ,  $j = 2, \dots, n$ , em relação ao nó  $n_0$ , tomado como referência.

## IV. 1

- 8 tensão:  $\phi \in \mathbb{R}^m$ , onde  $\phi_i$  representa a defasagem entre as tensões elétricas nas extremidades final e inicial do ramo  $r_i$ .
- 9 injeções de potência:  $P \in \mathbb{R}^n$ , onde  $P_j$  é a potência elétrica ativa injetada no nó  $n_j$ ,  $j = 1, 2, \dots, n$ . Note-se que, devido a (5ii), tem-se necessariamente

$$\sum_{j=0}^n P_j = 0 \quad \text{e portanto, a componente } P_0 \text{ pode ser obtida imediatamente a partir de } P.$$

- 10 Seja  $T$  uma matriz  $n \times m$ , obtida eliminando a primeira linha da matriz de incidência de  $(N, M)$  (linha correspondente ao nó  $n_0$ ).

Devido a A.16, é válida a relação:

$$11 \quad \phi = -T' \theta$$

- 12 Capacidade de um ramo: Considere-se um ramo  $r_i$ ,  $i = 1, 2, \dots, m$ , no estado  $S_i$ . Ao estado  $S_i$  corresponde (2) a composição do ramo  $r_i$  em termos de número de linhas de transmissão de cada tipo construídas no trecho  $r_i$ . Seja  $s$  o número de linhas em  $r_i$ . Define-se a capacidade do ramo  $r_i$  por

$$\gamma_i = \sum_{h=1}^s \frac{V_{ih}^2 X_{ih}}{Z_{ih}^2}$$

onde:  $V_{ih}$  : tensão elétrica nominal da  $h$ -ésima linha do ramo  $r_i$ .

$Z_{ih} = R_{ih} + jX_{ih}$  : impedância da  $h$ -ésima linha do ramo  $r_i$

Fluxo de carga :

Seja  $Q$  a matriz diagonal  $m \times m$  definida por

## IV.1

$$13 \quad Q_{ii} = \gamma_i \quad i = 1, 2, \dots, m$$

e seja  $A$  uma matriz  $n \times n$ , chamada daqui em diante de matriz de capacidades, definida por

$$14 \quad A = TQT' \quad \text{onde } T \text{ é a matriz (10).}$$

Sob as **hipóteses** (5), são **válidas** as seguintes relações entre **fluxos**, **defasagens** e **injeções**:

$$15 \quad A\theta = P$$

$$16 \quad f = Q\phi, \quad \text{ou seja, } f_i = \gamma_i \phi_i, \quad i = 1, 2, \dots, m$$

A relação (15) representa a equação do fluxo de carga linear e sua resolução fornece as fases nos **nós** de rede elétrica.

Um **critério** de segurança **será** enunciado a seguir, baseado no vetor  $\theta$ . Para garantir a existência e unicidade da solução de (15), é necessária a seguinte hipótese:

17 Hipótese: A rede parcial  $(N, M')$   $(N, M)$  tal que

$$M' = \{ r_i \in M \mid \gamma_i > 0 \} \quad \text{é conexa.}$$

Esta hipótese implica em que quaisquer dois **nós** do sistema de **potência** estão fisicamente ligados através de uma cadeia de linhas de **transmissão**.

Com a **hipótese** (17), garante-se a inversibilidade de  $A$ ,  $|10|$ , podendo-se escrever

$$18 \quad \theta = A^{-1} P \quad \text{e, usando (11),}$$

$$19 \quad \phi = T'A^{-1} P$$

## IV.1

- 20 critério de segurança : o seguinte critério de segurança é usual em estudos de sistemas de potência [12] .

Uma rede é considerada segura (**confiável**) se a perda de uma **linha**, **possivelmente** a maior, em qualquer ramo, não causar nenhum problema de fluxo excessivo ou instabilidade transitória que motive a perda de **equipamento** adicional.

A verificação da existência de fluxos **excessivos** é feita com base nas defasagens  $\phi$  . O procedimento normal [31] é estipular um valor **máximo, próximo** de  $36^\circ$  , para a defasagem permitida através de cada linha em qualquer situação de emergência, garantindo-se com isto a estabilidade **transitória** do sistema . Existe, além disso, um **limite térmico** para o fluxo em cada tipo de linha, que com a equação (16) fornece um limite de ângulo para cada ramo:  $\phi_i$  não deve ultrapassar, em nenhuma situação de emergência, o mínimo entre os **limites** de ângulo citados.

- 21 Neste trabalho supõe-se dado para cada configuração  $C$  um vetor  $\bar{\phi}(C) \in \mathbb{R}^m$  ,  $\bar{\phi}(C) > 0$  , chamado vetor de limites de defasagens para a configuração  $C$  , tal que, se o ramo  $r_i$  não tem nenhuma linha, então  $\bar{\phi}_i(C) = +\infty$  ,  $i = 1, 2, \dots, m$ .

Pode-se reenunciar o critério (20) com base no precedente. Chama-se de contingência no ramo  $r_i$  a retirada da linha de maior capacidade entre as linhas que compõem o ramo  $i$  . Com base no novo vetor de estados obtido a cada contingência, podem-se **recalcular** as capacidades  $\gamma_i$  e, portanto, as matrizes de capacidades.

Se nenhuma contingência provocar a violação da hipótese de conexidade (17) , as matrizes de capacidade são **inversíveis** e podem-se calcular os vetores de defasagens para cada contingência . Mantendo constante o vetor de injeções  $P$  (o que sempre é feito em otimização a curto prazo), introduz-se a notação a seguir, que será conveniente na próxima seção:

- 22 Seja  $C = (N, M, S)$  uma configuração e **suponha-se** que nenhuma contingência viola a hipótese (17). Seja  $\phi^0(c) \in \mathbb{R}^m$  o

## IV.1

vetor de defasagens correspondente a  $C$  com as injeções  $P$  fixadas calculado por (19), e seja  $\bar{\phi}(C)$  o vetor de limites de defasagens (21).

23 Considere-se a contingência no ramo  $r_i$ ,  $i = 1, 2, \dots, m$ . Definem-se os vetores  $\phi^i(C) \in R^m$ ,  $i = 1, 2, \dots, m$ , por

(i) Se  $r_i$  não tem nenhuma linha, então  $\phi^i(C) = 0$

(ii) Se  $r_i$  tem linhas,  $\phi^i(C)$  é o vetor de defasagens obtido para  $C$ , retirando de  $r_i$  a linha de maior capacidade, com a imposição abaixo:

(iii) Se  $r_i$  tem uma linha somente, então  $\phi^i(C) = 0$

24 Definição: Define-se o vetor de defasagens máximas para  $C$ ,  $\hat{\phi}(C) \in R^m$ , por:

$$\hat{\phi}_i(C) = \phi_i^q(C), \quad \text{onde } q \text{ é tal que}$$

$$|\phi_i^q(C)| = \max_{p=0, 1, \dots, m} |\phi_i^p(C)|$$

Com esses elementos, pode-se enunciar um critério de segurança, capaz de garantir as **condições** de segurança impostas em (20), dentro da precisão do modelo adotado:

25 Critério de Segurança (viabilidade): Uma configuração  $C$  é considerada segura se

(i) Nenhuma contingência viola a condição de conexidade (17) e, além disso,

(ii)  $|\hat{\phi}_i(C)| \leq \bar{\phi}_i(C) \quad i = 1, 2, \dots, m$

O critério acima pode ser imediatamente colocado no

## IV.1

**formato 11.1.11** , mas isso não **será** feito porque não traz nenhuma vantagem. O critério (25) é bastante complexo e sua verificação é chamada Análise de Contingências , constituindo na prática uma rotina computacional separada, à qual **será dedicada a próxima** seção deste trabalho,

## Seção 2 - Análise de Contingências

A análise de contingências consiste na verificação do critério (25). Para cada contingência, pode-se recalcular a matriz de capacidades e invertê-la (se for **possível**), calculando as defasagens.

O procedimento acima **é** muito trabalhoso e não utiliza o fato de poucos elementos da matriz de capacidades  $A$  (14) serem **modificados** em cada contingência. Um estudo da sensibilidade de  $\phi$  em relação à variação de capacidade em um ramo tem sido **utilizado** em otimização a curto prazo [9], [10], através do cálculo da matriz jacobiana  $\partial\phi/\partial\gamma$  . No entanto, a verificação do critério deve ser mais precisa do que se pode obter desta forma, o que motiva o **emprego** de um método direto de cálculo dos vetores  $\phi^i(C)$  .

Um método sem inversões de matrizes para esse cálculo foi desenvolvido em [32] e seus resultados são **listados** a seguir. Parte-se de uma análise de sensibilidades, em que a variação de fases  $\Delta\theta$  devido à variação  $\Delta\gamma_p$  da capacidade de um ramo  $r_p$  **é** calculada sem aproximação.

## 26 Análise de sensibilidades

Seja dada uma **configuração**  $C = (N, M, S)$  , de acordo com a seção anterior. Note-se que os ramos são orientados segundo a ordem crescente de seus **nós** terminais. Suponham-se conhecidas as injeções  $P \in \mathbb{R}^n$  e a matriz  $A^{-1}$  , calculada segundo (14) para a configuração  $C$  .

## IV.2

Suponha-se que o ramo  $r_p = (n_r, n_s)$  sofre uma variação  $\Delta\gamma_p$  de capacidade. A variação correspondente nas fases dos nós de  $C$  serão dadas pelos resultados abaixo.

- 27 Se a variação de capacidade não viola a condição de conexidade (17), então

$$\Delta\theta^p = \alpha (A_s^{-1} - A_r^{-1}) \phi_p, \text{ onde}$$

$$\alpha = \frac{-\Delta\gamma_p}{1 + \Delta\gamma_p (A_{rr}^{-1} + A_{ss}^{-1} - 2A_{rs}^{-1})}$$

Com as convenções:

$$A_j^{-1} : j\text{-ésima coluna de } A^{-1} ; j \neq 0 ; A_0^{-1} = 0$$

$$A_{jk}^{-1} : \text{elemento da } j\text{-ésima linha, } k\text{-ésima coluna de } A^{-1}$$

$$A_{oj}^{-1} = A_{jo}^{-1} = \theta_o = 0, \quad j = 0, 1, \dots, n$$

- 28 A condição de conexidade (17) é violada se e somente se

$$1 + \Delta\gamma_p (A_{rr}^{-1} + A_{ss}^{-1} - 2A_{rs}^{-1}) = 0$$

Se a contingência do ramo  $r_i$ ,  $i = 1, 2, \dots, m$  não violar (17), pode-se escrever, usando (11),

- 29  $\phi^i(C) = \phi - T' \Delta\theta^i$ , sempre que forem satisfeitas as condições em (23 ii)

onde  $\phi = \phi^0(C)$  e  $\Delta\theta^i$  é calculado por (27).

- 30 Com esses elementos, pode-se estabelecer uma rotina de teste do critério (25): o critério de segurança é satisfeito se, para cada  $i = 1, 2, \dots, m$ , a variação de capacidade  $\Delta\gamma_i < 0$  devida à retirada da linha de maior capacidade (se houver alguma) do ramo  $r_i$



## IV.2

satisfizer

$$(i) \quad 1 + \Delta\gamma_i (A_{rr}^{-1} + A_{ss}^{-1} - 2A_{rs}^{-1}) \neq 0, \text{ com } r_i = (n_r, n_s)$$

além disso deve-se satisfazer

$$(ii) \quad |\hat{\phi}_i(c)| \leq \bar{\phi}_i(c), \quad i = 1, 2, \dots, m, \text{ onde } \hat{\phi}(c) \quad (24) \text{ é calculado usando (29) e (23).}$$

### 31 Modificações da matriz $A^{-1}$

Embora a **análise** de contingência seja executada diretamente por (29), é vantajoso para a **sequência** de trabalho obter expressões para o **cálculo** das matrizes  $(A^i)^{-1}$  correspondentes à matriz de capacidades para a configuração obtida a partir de  $C$  por meio de uma variação de capacidade do ramo  $r_i$ . As expressões abaixo estão desenvolvidas em [32]

Suponha-se, como anteriormente, que o ramo  $r_p = (n_r, n_s)$  sofre uma variação de capacidade  $\Delta\gamma_p$ . Neste caso, se não for violada a condição de conexidade,

$$32 \quad (A^p)^{-1} = A^{-1} + \alpha(A_s^{-1} - A_r^{-1})(A_s^{-1} - A_r^{-1}); \quad ,$$

com as convenções e notação em (27).

A expressão (32) permitirá que em toda a **otimização**, a curto e a longo prazo, somente uma inversão de matriz seja **necessária**, se a topologia não variar de estágio a estágio.

Durante a aplicação do **algoritmo** de expansão 111.2.30 a redes de transmissão de **potência** não é **necessário**, portanto, fazer inversões de matrizes, desde que se possua a inversa da matriz de capacidades de  $C^0$ . Cada vez que o passo 1 pesquisa uma **nova opção**, obtém-se a inversa da matriz de capacidades para essa **configu-**

## IV.2

ração, a partir de sua genitora no grafo de expansão, usando (32).

### Seção 3 - Cálculo das perdas

O cômputo do custo da operação para uma configuração , segundo a expressão 11.2.26 pode levar em conta manutenção, perdas de energia e o custo da energia de substituição nas horas de ponta. Não se consideram neste trabalho os custos de manutenção, e apresentam-se a seguir as expressões para o cálculo das perdas.

Para a aproximação linear do fluxo de carga, as perdas instantâneas de energia no ramo  $r_i$  da configuração introduzida em (5) são calculadas em [10]: pelas expressões abaixo, onde  $s$  é o número de linhas presentes em  $r_i$ ,  $V_{ip}$  é a tensão nominal da primeira linha de  $r_i$ ,  $Z_{ip} = R_{ip} + jX_{ip}$  sua impedância.

$$33 \quad L_i = \frac{1}{2} \sum_{p=1}^s \frac{V_{ip}^2 R_{ip}}{Z_{ip}^2} \phi_i^2$$

As perdas instantâneas na configuração são dadas por

$$34 \quad L = \sum_{i=1}^m L_i$$

O cálculo do custo das perdas durante um estágio é feito normalmente usando um fator de carga médio para o sistema e depende do problema particular em tratamento. Para o sistema Sudeste, a metodologia de computo das perdas está exposto em [24] e será apresentada no capítulo VII. Na próxima seção supõe-se conhecida essa metodologia.

## IV.4

Seção 4 - EXPANSÃO DE REDES DE TRANSMISSÃO DE ENERGIA ELÉTRICA

O algoritmo 111.2.30 pode ser aplicado à expansão de redes de transmissão de energia elétrica, desde que os custos de implementação sejam sempre positivos. Resta somente estabelecer um método para a construção das listas de opções de maneira eficiente, segundo as observações (111.2.40), o que constitui o objetivo desta seção. Primeiramente, porém, evidenciam-se as hipóteses de trabalho e introduzem-se algumas simplificações (não essenciais), com o intuito de manter a abordagem estreitamente ligada à prática de projeto. Resultados da aplicação do algoritmo encontram-se no Capítulo VII. Usa-se a notação introduzida nos capítulos anteriores.

- 35 Hipótese: Somente são permitidas transições de estado elementares que aumentem a capacidade de um ramo. Resulta que os custos de transição de estados são sempre positivos, satisfazendo-se a hipótese (111.1.14).

Em termos de linhas de transmissão, pode-se colocar esta hipótese no seguinte formato, usual na prática:

- 36 As transições de estado elementares correspondem à adição de uma linha a um ramo, ou à substituição de uma linha por outra de maior capacidade. Esta segunda alternativa aplica-se a sistemas em que restrições geográficas não permitem o aumento do número de linhas, não ocorrendo nos sistemas brasileiros.

- 37 Hipótese: A economia no custo das perdas resultante de uma transição de estado é inferior ao custo da transição de estado. Esta

hipótese corresponde a III.1.11 e é satisfeita com larga margem nos casos estudados (flutuações do custo das perdas dificilmente ultrapassam 5% do custo de implementação correspondente).

As hipóteses acima são suficientes para a aplicabilidade do algoritmo 111.2.30. Formula-se a seguir uma hipótese de "convexidade" para os conjuntos de estados sucessores  $\delta(s^0, t)$ . Esta hipótese é desnecessária, mas é satisfeita sempre nos casos práticos e traz alguma simplificação computacional. Segue-se a notação do capítulo anterior.

- 38 Hipótese: Sejam  $c^0, c^1, c^2$  configurações com vetores de estado  $s^1, s^2, s^3$ , respectivamente. Nesse caso, se  $s^1, s^2 \in \delta(s^0, t)$ , então os estados correspondentes a qualquer caminho entre  $c^1$  e  $c^2$  estão em  $\delta(s^0, t)$ .

Em outras palavras, se um caminho no grafo de expansão  $H$  sai do conjunto  $\delta(s^0, t)$ , não pode voltar a penetrar nesse conjunto. Conjuntos de estados sucessores com essas características são descritos em VI.1.9, e correspondem aos conjuntos utilizados nos exemplos do Capítulo VII.

A hipótese (38) permite algumas simplificações no algoritmo.

- 39 Simplificação do algoritmo de expansão: Na construção da lista de opções para uma configuração  $c^n$  tal que  $s^n \in \delta(s^0, t)$ , consideram-se somente elementos de  $\Lambda(c^n)$  com estado em  $\delta(s^0, t)$ .

Se  $c^0 \in \delta(s^0, t)$ , essa simplificação pode sempre ser feita e o passo 2 deve apenas verificar a viabilidade das confi

## IV.4

gurações a serem fechadas.

#### 41 Observação : Lista Fechado

Quando somente adições de linhas à rede são permitidas, resulta que qualquer caminho entre dois nós do grafo de configurações tem o mesmo comprimento e mesmo custo, correspondentes respectivamente ao número e custo das linhas que devem ser construídas. Neste caso pode-se particionar a lista Fechado segundo os níveis e dispensa-se a estocagem dos custos, obtendo-se economia em tempo de computação e memória. Quando substituições de linhas são também permitidas, podem-se fazer simplificações semelhantes, dependentes do projeto particular,

As considerações acima são suficientes para a aplicação do algoritmo, desde que se disponha de um método de ordenação e truncagem das listas de opções, objeto do restante deste capítulo. Aspectos computacionais relacionados ao algoritmo serão discutidos no Capítulo VII.

#### 42 construção das listas de opções

Os elementos das listas de opções são configurações  $C$  resultantes de transições elementares de estado  $(S^n, S)$ , sendo a descrição completa de  $C^n$  acessível. Em termos computacionais, é suficiente guardar o índice do ramo afetado pela transição e o tipo de linha a ser construído ou o tipo de substituição de linha. Essas listas devem ser ordenadas por um critério de eficiência, como se observou em 111.2.40.

## IV.4

Um primeiro critério para sua ordenação é o custo das transições: trata-se de um critério de pouca valia, uma vez que não há correlação entre o custo de uma linha e sua eficiência no sentido de tornar a rede viável.

Uma boa ordenação deve seguir um critério baseado não somente em custos, como também no efeito de cada transição sobre a segurança do sistema. Um critério desse tipo foi desenvolvido em [9], com a exigência de diferenciabilidade. Desenvolve-se, a seguir, um novo critério de ordenação sem essa exigência.

Segue-se a notação do capítulo anterior, sendo fixos o estágio  $t$  e as injeções de potência  $P$ . Dispõe-se de uma configuração  $C^n = (N, M, S^n)$  não viável gerada pelo algoritmo e procura-se ordenar a lista de opções construída pelo passo 3 do algoritmo.

Supõe-se que a condição de conexidade (30i) é satisfeita em todas as contingências. Desconexões serão comentadas abaixo, em (48).

43 O carregamento presente em uma configuração  $C$  é caracterizado pelo vetor de defasagens máximas  $\hat{\phi}(C)$  definido em (24). Procura-se associar a cada configuração  $C$  um número real  $r(C)$  que caracterize sua "eficiência", no sentido de que  $r(C)$  deve ser tanto menor quanto menos custoso seja viabilizar  $C$  por meio de aumentos de capacidade. O método de cálculo de  $r$  é essencialmente heurístico e deve ter as seguintes características:

- (i)  $r(C)$  deve depender das defasagens máximas  $\hat{\phi}_i(C)$ ,  $i=1,2,\dots,m$  crescendo rapidamente com os valores de  $\hat{\phi}_i(C)$ .
- (ii)  $r(C)$  deve penalizar os ramos sobrecarregados, isto é, deve

## IV.4

ser muito afetado por defasagens tais que  $|\hat{\phi}_i(C)| \geq \bar{\phi}_i(C)$   
(Ver 25) .

(ii) A influência de cada defasagem  $\hat{\phi}_i(C)$  sobre  $r(C)$  deve ser tanto maior quanto mais alto for o custo do reforço do ramo  $i$  (considera-se o custo da transição de menor custo no ramo  $i$  . Pode-se também considerar o comprimento do ramo).

Segue-se um formato proposto para o cálculo de  $r$  , utilizado ,com bons resultados na implementação descrita no Capítulo VII .

44 Definição: Dada uma configuração  $C$  que não viola (30i) em nenhuma contigência, define-se o coeficiente de carregamento de  $C$  por meio de

$$r(C) = \sum_{i=1}^m \beta_i h_i$$

onde, para  $i=1,2,\dots,m$  ,

$$h_i = \begin{cases} (\hat{\phi}_i(C) / \bar{\phi}_i(C))^2 & \text{se } |\hat{\phi}_i(C)| < \bar{\phi}_i(C) \\ \alpha (\hat{\phi}_i(C) / \bar{\phi}_i(C))^2 & \text{se } |\hat{\phi}_i(C)| \geq \bar{\phi}_i(C) \end{cases} , \text{ com } \alpha > 1$$

$\beta_i$  é um real positivo que cresce com o custo de uma transição de estados de custo mínimo no ramo  $r_i$  , ou, alternativamente,  $\beta_i$  cresce com o comprimento do caminho correspondente a  $r_i$  .

Na implementação do Capítulo VII, usa-se  $\alpha=10$  , e  $\beta_i$  é igual ao custo de uma linha de custo mínimo para o ramo  $r_i$  .



## IV.4

A ordenação das opções será feita por um critério de eficiência das transições de estado. O critério será tanto maior quanto mais "eficiente" for uma transição de estados: deve crescer com a variação (para menor) do coeficiente de carregamento de vida à transição, decrescendo com o custo da transição. Segue-se o formato proposto para o cálculo do critério.

Seja  $C \in \Lambda(C^n)$ .

- 45 Definição : O critério de eficiência  $g(C^n, C)$  correspondente ao ramo  $(C^n, C)$  de  $H$  é definido por

$$g(C^n, C) = \frac{r(C^n) - r(C)}{c_t(C^n, C, t)}$$

- 46 ordenação das listas de opções: Ordena-se as listas de opções segundo a ordem decrescente dos critérios de eficiência (45) correspondentes a seus elementos. Para isto, calculam-se os valores do critério segundo (44).

- 47 Observação: A definição do critério de eficiência depende de fatores heurísticos. Além disso, o cálculo do critério será agroximado por meio de estudo de sensibilidades. Observe-se no entanto que a Única função do critério até o presente é aumentar a eficiência do algoritmo e, portanto, nada se perde pela aplicação de heurística no que diz respeito aos resultados. O efeito das aproximações pode ser sentido somente como resultado da truncagem das listas, comentada adiante.

- 48 Tratamento das desconexões

Considere-se que a contigência em um ramo  $r_i$  de uma

## IV.4

configuração  $C^n$  viola a condição de conexidade (30i). Pretende-se adaptar o cálculo do critério de eficiência (45), de modo que transições  $(C^n, C)$  que eliminem as desconexões forneçam um valor elevado de  $g(C^n, C)$ .

É suficiente para isso calcular  $\hat{\phi}(C)$  como se houvesse uma grande sobrearga no ramo  $r_i$  que provoca a desconexão, fazendo, por exemplo,

$$49 \quad \hat{\phi}_i(C^n) = 100 \phi_i^0(C^n),$$

ignorando o vetor  $\phi^i(C^n)$  no cálculo de  $\hat{\phi}(C^n)$ , em (24).

Obter-se-á evidentemente um valor grande de  $g(C^n, C)$  associado às configurações  $C$  em que alguma desconexão for eliminada.

#### 50 Cálculo do critério de eficiência

O cálculo dos valores de  $g(C^n, C)$  em (45) dependem da avaliação dos valores de  $r(C)$  para cada configuração em  $\Lambda(C^n)$ , e portanto dos vetores  $\hat{\phi}(C)$  para essas configurações. Um cálculo aproximado desses vetores será feito usando a análise de sensibilidade exposta em (26).

Seja  $C$  um elemento da lista de opções gerada por  $C^n$ . Como  $(S^n, S)$  é uma transição elementar, somente uma componente do vetor  $\gamma(S) - \gamma(S^n)$  é diferente de zero, onde  $\gamma$  é o mapeamento que fornece os vetores de capacidade definido em II.1.7.

$$51 \quad \text{Seja } p \text{ o índice dessa componente e seja } \Delta\gamma_p = \gamma_p(S) - \gamma_p(S^n). \text{ Usando-se as expressões (27) e (11), calcula-se o valor de } \Delta\phi^p \in R^m, \text{ tal que } \phi^0(C) = \phi^0(C^n) + \Delta\phi^p.$$

Deseja-se porém calcular  $\phi(C)$ . Uma aproximação para  $\hat{\phi}(C)$  é obtida considerando-se que a variação de  $\Delta\phi^P$  é pequena para pequenas variações no vetor de capacidade  $\gamma(C)$ , desde que nenhuma contingência de  $C^n$  provoque desconexão, como se comenta em [32]: nesse caso, pode-se escrever

$$\hat{\phi}(C) - \hat{\phi}(C^n) \approx \phi^O(C) - \phi^O(C^n) \quad \text{e portanto}$$

$$52 \quad \hat{\phi}(C) \approx \hat{\phi}(C^n) + \Delta\phi^P$$

Podem-se usar métodos mais precisos para o cálculo de  $\hat{\phi}(C)$  [30], mas isso leva a um aumento nos requisitos de memória e tempo de computação.

### 53 Tratamento das desconexões:

O cálculo de  $A$  acima não pode prever a ocorrência de desconexões na análise de contingências de  $C^n$ . No entanto, o procedimento acima resolve o problema de eliminação de desconexões, devido ao seguinte:

54 Se a contingência em um ramo  $r_i$  provoca a desconexão da rede, o fluxo de potência passando por  $r_i$  em  $C^n$  é dado pela diferença entre as somas das injeções em cada componente conexa de  $(N,M)$  obtida com a retirada de  $r_i$ , pois  $\sum_{j=0}^n P_i = 0$ .

Seja  $r_p$  um ramo qualquer de  $(N,M)$ , e seja  $\Delta\phi_i^P$  a variação da defasagem  $\phi_i^O(C^n)$  no ramo  $r_i$ , devida ao reforço do ramo  $r_p$ , calculada em (51).

Se a construção de uma linha em  $r_p$  não elimina a desconexão provocada por  $r_i$ , então  $\Delta\phi_i^P = 0$ . De fato, o fluxo  $f_i$

## IV.4

em  $r_i$  será dado por (54) e  $\phi_i^0(C) = f_i/\gamma_i$  devido a (16), com ou sem o reforço de  $r_p$  e portanto  $\Delta\phi_i^p = 0$ .

Se o reforço de  $r_p$  elimina a desconexão, então o fluxo em (54) será dividido pelos ramos  $r_i$  e  $r_p$  após o reforço, provocando  $\Delta\phi_i^p \neq 0$ . Devido a (49),

$$\bar{\phi}_i(C^n) = 100 \phi_i^0(C^n)$$

e, devido a (52),

$$\bar{\phi}_i(C) = 100 \phi_i^0(C^n) + \Delta\phi_i^p$$

Observando-se as expressões usadas no cálculo de  $g(C^n, C)$ , verifica-se que, mesmo para pequenos valores de  $\Delta\phi_i^p$ , será grande o valor de  $g(C^n, C)$ , devido ao fator

$$\frac{2\alpha \cdot 100 \cdot \phi_i^0(C^n) \cdot \Delta\phi_i^p}{(\bar{\phi}_i(C^n))^2}, \text{ resultante do cálculo de } r(C^n) - r(C).$$

55 Observação: A desconexão não será detetada sempre que

$100 \phi_i^0(C^n) < \bar{\phi}_i(C^n)$ , o que não restringe o método proposto: não há interesse em evitar desconexões provocadas por ramos cujo carregamento é muito pequeno. Pode-se sempre usar um multiplicador maior do que 100 ( $10^6$ , por exemplo), se desconexões forem sempre indesejáveis. Comenta-se no Capítulo VII um caso em que uma desconexão não deve ser evitada.

Completou-se o estudo da ordenação das listas de opções e até o presente não se modificaram os resultados da aplicação do algoritmo de expansão, permanecendo válida a tese do teorema III. 2.39. Para problemas de grande porte, o tempo gasto na exploração

de opções ineficientes torna-se proibitivo e deve-se limitar a busca por um procedimento heurístico: a truncagem das listas de opções.

- 56 Truncagem das listas de opções : As listas de opções obtidas segundo (42) , ordenadas por ordem decrescente do critério de eficiência  $g(c^n, c)$  , podem ser truncadas por vários critérios. Introduz-se assim uma limitação heurística ao número de configurações testadas pelo algoritmo. Resultados excelentes foram obtidos com os seguintes critérios:
- (i) Eliminam-se as opções  $c$  tais que  $g(c^n, c) < \lambda \bar{g}$  , onde  $g$  é o maior valor de  $g(c^n, c)$  para  $c$  pertencente à lista de opções,  $\lambda < 1$  é um número inteiro. Usou-se  $\lambda = 0,1$  nos exemplos apresentados no Capítulo VII .
  - (ii) Mantém-se na lista somente  $q$  elementos, onde  $q$  é calculado por  $q = Q + n - \sum_{i=1}^n o_i$  ,  $Q$  é um valor fixo. Desta forma, limita-se a  $Q$  o "desvio" em relação ao primeiro caminho encontrado, de modo que configurações resultantes de opções de baixa ordem tentam mais sucessores no grafo  $H$  que configurações resultantes de opções mais pobres.

As considerações acima concluem o estudo de otimização a curto prazo. Nos capítulos seguintes, parte-se do pressuposto de que existe uma rotina capaz de calcular os conjuntos de sucessoras eficientes de uma configuração  $c^n$ ,  $\hat{r}(c^n)$  ou  $\hat{r}_k(c^n)$ . Resultados computacionais encontram-se no Capítulo VII .

## CAPÍTULO V

### PROGRAMAÇÃO HEURÍSTICA E PROCESSOS DE DECISÕES SEQUENCIAIS

Este capítulo trata de algumas extensões do algoritmo  $A^*$  desenvolvido em [6], de modo a adaptá-lo às resoluções de problemas de decisões sequenciais de grande porte e, em particular, à resolução do Problema de Planejamento a Longo Prazo proposto no Capítulo II. O tratamento, no entanto, é geral e sua aplicação a Planejamento será feita no capítulo seguinte.

Algoritmos de Programação Heurística desenvolveram-se no contexto de Inteligência Artificial, recebendo um tratamento formal em [6] e sendo posteriormente estudados em [2], [7], [8]. Embora as técnicas obtidas sejam muito poderosas, sua utilização parece ter-se confinado ao domínio em que foram concebidas, não existindo nenhum tratamento geral de sua aplicação a problemas de decisões sequenciais.

Parte-se neste trabalho do algoritmo  $A^*$ , destinado à busca de caminhos em grafos, cuja descrição encontra-se no Apêndice B. A eficiência de  $A^*$  depende de uma função avaliação obtida por critérios heurísticos. Dentro de certas condições, garante-se que  $A^*$  resolve o problema de busca de caminhos de custo mínimo; em outros casos, podem-se obter rapidamente caminhos cujo custo não é necessariamente mínimo. O algoritmo  $\hat{A}$  proposto neste trabalho explora este último fato, procurando, através de modificações iterativas da função avaliação, obter uma sequência de soluções para

V.

o problema de busca, de modo que soluções sub-Ótimas sejam disponíveis rapidamente. Dentro de certas condições, essa sequência tende, em tempo finito, para uma solução Ótima.

Esse comportamento torna-se extremamente vantajoso em problemas de grande porte, em que a obtenção de soluções Ótimas é **inviável** computacionalmente por qualquer método conhecido até o presente.

Na seção 2, estende-se o **algoritmo  $\hat{A}$**  a problemas de busca em grafos cujos nós são parcialmente ordenados por uma relação de preferência usual em problemas de planejamento, obtendo-se alguma simplificação computacional. A seção 3 trata de uma abordagem de grafosa uma classe de problemas de decisões sequenciais, comentando-se sua resolução por meio do **algoritmo  $\hat{A}$**  proposto.

O **conteúdo** deste **capítulo** é independente do restante da tese, sendo utilizado na **sequência** para resolver o problema PLP proposto em 11.3.36 . A notação usada é semelhante à de  $|2|$ , sendo introduzida na seção 1 .

Seção 1 - EXTENSÃO DO ALGORITMO A\*

1            Seja  $N$  um conjunto de nós e  $\Gamma : N \rightarrow P(N)$  um opera  
dor. Considere-se o grafo  $H = (N, \Gamma)$  e seja  $M = \{(n, n') \mid n \in N, n' \in \Gamma(n)\}$  o conjunto de ramos de  $H$  (os conceitos de teoria de grafos estão resumidos no Apêndice A).

2            Seja  $c : M \rightarrow R$  uma função custo que associa a cada ramo  $r = (n, n')$  de  $H$  um custo  $c(r) = c(n, n')$ . Dado um caminho  $G = (n_1, n_2, \dots, n_p)$  em  $H$ , associa-se a  $G$  o custo

$$3 \quad c(G) = c(n_1, n_2, \dots, n_p) = \sum_{i=2}^p c(n_{i-1}, n_i) .$$

4 Definição : Sejam  $n, n' \in N$  . Define-se a aplicação

$$h : N \times N \rightarrow [-\infty, +\infty] \text{ por}$$

$$h(n, n') = \begin{cases} \inf \{c(G) \mid G \text{ é um caminho de } n \text{ para } n'\}, & \text{se } n' \neq n \\ 0, & \text{se } n' = n \end{cases}$$

Usa-se a convenção:

$$\inf \emptyset = +\infty$$

5 Definição : Sejam  $S, T \subset N$  . Define-se

$$h(S, T) = \inf \{h(n, n') \mid n \in S, n' \in T\}$$

6 Lema : Se  $H$  é finito e não possui circuitos de custo negativo, então

$$(\forall n, n' \in N) \quad h(n, n') > -\infty$$

Demonstração : Ver B . 8



V.1

Sejam  $S, T \subset N$  conjuntos denominados, respectivamente, conjunto inicial e alvo. A confusão de notação existente nas definições a seguir, bem como nas definições de custos (3) é usual e inócua: diferencia-se a aplicação de  $c$  ou  $h$  por seus argumentos.

7 Definição: Dado  $n \in N$ , define-se

$$h(n) \triangleq h(n, T)$$

Dado  $A \subset N$ , define-se

$$h(A) \triangleq h(A, T)$$

8 Definição: Dado  $n \in N$ , define-se

$$g(n) = h(S, n)$$

Com esses elementos, pode-se introduzir o problema de busca:

Problema de Busca de Caminhos de Custo Mínimo (PB) :

Considere-se um grafo  $H = (N, E)$  e sejam  $S, T \subset N$ .

Encontrar nós  $s \in S$ ,  $t \in T$  e um caminho  $G$  de  $s$  para  $t$ , tais que

$$c(G) = h(S)$$

Passa-se à apresentação de um formato modificado do algoritmo  $A^*$ , cujas **características** serão exploradas em seguida. Vários algoritmos podem ser usados na resolução de (PB), sendo comentados em [4], e a admissibilidade de cada um é garantida sob

certas hipóteses (um **algoritmo** é admissível se garantir a resolução de PB sempre que exista solução). As hipóteses mais comuns nas demonstrações de admissibilidade são:

Grafos finitos : Impõe-se a não-negatividade dos custos de todos os ramos, como no **algoritmo** de Dijkstra, ou impõe-se a **inexistência** de circuitos de custo negativo, como em [33].

Grafos infinitos : Usa-se a hipótese de que todos os custos são superiores a uma constante  $\delta > 0$ , o que define  $\delta$ -grafos. e é usado no **algoritmo**  $A^*$  [6].

A demonstração da admissibilidade de  $A^*$  para grafos finitos sem circuitos de custo negativo é imediata, podendo-se concluir que, na realidade, o **algoritmo** proposto em [33] não passa de um caso particular de  $A^*$ .

Neste trabalho usam-se essas últimas hipóteses, por conveniência. A extensão dos resultados obtidos para  $\delta$ -grafos infinitos é imediata.

## 10 O Algoritmo $\hat{A}$

O **algoritmo** manipula duas listas, cujos elementos são nós  $n_i$  de  $H$ , acompanhados de custos  $\hat{g}(n_i)$ ,  $\hat{h}(n_i)$  definidos pelo **algoritmo** e de apontadores, também definidos pelo **algoritmo**. As listas são:

11 Lista Aberto, cujos elementos são chamados nós abertos ;

12 Lista Fechado, cujos elementos são chamados nós fechados .

V.1

Inicialmente ambas as listas estão vazias. São dados os conjuntos  $S, T$  e demais elementos definidos acima.

13 Algoritmo:  $\hat{A} : S, T \subset N$  ,  $S$  finito

Passo 0 :

Defina a função  $\hat{h} : N \rightarrow R$  , introduza na lista **Aberto** os nós de  $S$ , associando a cada  $n \in S$  os valores  $\hat{g}(n)=0$  ,  $\hat{h}(n)$ .

Passo 1: :

Se a lista aberto estiver vazia, pare

Senão, escolha entre os nós abertos um nó  $\bar{n}$  tal que

$$f(\bar{n}) = \min\{f(n) | n \text{ aberto}\} , \text{ onde } f(n) = \hat{g}(n) + \hat{h}(n)$$

Resolva empates arbitrariamente, mas dando preferência a nós  $n \in T$

Passo 2 :

Se  $\bar{n} \in T$ , vá para o passo 5

Senão, transfira  $\bar{n}$  para a lista Fechado

Passo 3 :

Obtenha  $\Gamma(\bar{n})$ , lista de sucessores de  $\bar{n}$

$$\text{Faça } \hat{g}(n) = \hat{g}(\bar{n}) + c(\bar{n}, n) \quad (\forall n \in \Gamma(\bar{n}))$$

Compare cada sucessor com todos os elementos das listas Aberto e Fechado

Retire de  $\Gamma(\bar{n})$  todos os elementos  $n$  eliminados pela regra

(14) por algum nó listado em Aberto ou Fechado, ou por algum

elemento de  $\Gamma(\bar{n})$  de índice inferior ao de  $n$  na lista  $\Gamma(\bar{n})$ \*

---

\* A comparação com outros elementos de  $\Gamma(\bar{n})$  tem importância em grafos parcialmente ordenados, discutidos na seção seguinte. Devem-se eliminar sucessores "ineficientes".

V.1

Retire da lista Aberto todos os **nós** eliminados pela regra (14) por **algum nó** remanescente em  $r(\bar{n})$

Introduza na lista Aberto os **nós** remanescentes em  $r(\bar{n})$ , associando-lhes o indicador  $\bar{n}$

Passo 4 :

Redefina a função  $\hat{h} : N \rightarrow R$ , atualize seus valores nas listas Aberto e Fechado. Vá para o passo 1

Passo 5 :

Se algum critério de parada for satisfeito, (Ver 22), termine o algoritmo, recuperando um caminho entre  $S$  e  $\bar{n}$  por meio dos indicadores, de **trás** para frente

Senão, vá para o passo 4

#### 14 Regra de eliminação :

$$n \text{ elimina } n' \text{ se } n=n' \text{ e } \hat{g}(n) \leq \hat{g}(n')$$

#### 15 Observações :

O algoritmo  $\hat{A}$  é equivalente ao  $A^*$ , se suprimirem-se os passos 4 e 5, terminando o algoritmo ao escolher  $\bar{n} \in T$  no passo 1. Define-se  $\hat{h}$  como uma função, o que não é necessário em  $A^*$ , não sendo também necessário em  $\hat{A}$  : o valor de  $\hat{h}(n)$  pode depender da maneira pela qual  $n$  foi obtido. Tal caso é pouco comum e dificulta a notação, podendo-se encontrar um exemplo em [34].

**Nós** fechados eliminados por (14) no passo 3 não são retirados da lista Fechado por razões computacionais comentadas adiante. Sua retirada em nada modifica a atuação do algoritmo.

V.1

A regra de eliminação foi colocada em evidência devido à sua utilidade na seção 2 .

A hipótese a seguir **será** utilizada no estudo de admissibilidade de  $\hat{A}$  , como se comentou acima.

- 16 **Hipótese** :  $H$  é finito e não possui circuitos de custo negativo.

Estuda-se a seguir o que se pode esperar do **algoritmo**  $\hat{A}$  . Note-se que o **algoritmo** obtém um caminho entre  $S$  e  $T$  sempre que o passo 5 é atingido.

- 17 **Lema** : Suponha-se que  $H$  satisfaz (16) e  $T$  é **acessível** a partir de  $S$  . Nesse caso, qualquer que seja a atuação do passo 4, o **algoritmo** atinge o passo 5 em um número finito de **iterações**.

**Demonstração** : Por absurdo: se o **algoritmo** não atinge o passo 5 , há duas hipóteses:

- (a) o **algoritmo** parou no passo 1, com a lista Aberto vazia .

Seja  $P = (n_1, n_2, \dots, n_p)$  um caminho qualquer com  $n_1 \in S$  ,  $n_p \in T$  .

Seja  $n_k$  o **nó** de  $P$  de mais alto Índice presente na lista **Fechado** no momento da parada ,  $k < p$  ( $n_k$  existe pois  $n_1$  entrou na lista Aberto no passo 1 e o passo 3 somente modifica os custos associados a **nós** abertos) .

Como  $n_k$  **está** fechado, em alguma **iteração** gerou-se  $n_{k+1} \in T(n_k)$ .  $n_{k+1}$  deve estar em Fechado, uma vez que Aberto **está** vazia, o que é absurdo.

V.1

(b) O algoritmo **cicla** indefinidamente entre os passos 1 e 4, sem encontrar  $n$  e  $T$ .

Como  $N$  é finito e em cada **iteração** um elemento é retirado de Aberto, existe um **nó**  $\bar{n}$  que entra na lista Aberto um **número in-**  
finito de vezes.

Cada vez que  $\bar{n}$  entra em Aberto,  $\hat{g}(\bar{n})$  corresponde ao custo de um caminho  $C$  entre  $S$  e  $\bar{n}$ , por construção.  $C$  não contém circuitos, pois se  $C = (n_1, n_2, \dots, n_k, n_{k+1}, \dots, \bar{n})$  com  $n_k = \bar{n}$ , então  $\bar{n} = n_k$  já estaria em alguma lista com  $\hat{g}(n_k) = c(n_1, n_2, \dots, n_k) \leq c(C)$  (por hipótese), o que acarretaria a **eliminação** de  $\bar{n}$  da lista de sucessores pela regra (14).

Cada vez que  $\bar{n}$  entra em Aberto,  $\hat{g}(\bar{n})$  é reduzido, pois a nova instância não é eliminada pela regra (14): cada **instância** corresponde, portanto, a um caminho diferente.

Existe portanto, um **número** infinito de caminhos sem circuitos em  $H$ , o que é absurdo para  $N$  finito.

(a) e (b) demonstram o lema.

18 **Corolário** : Se  $H$  satisfaz (16), então o **algoritmo pára** no passo 1 sem ter atingido o passo 5 se e somente se  $T$  não for acessível a partir de  $S$ .

**Demonstração** :

Se  $T$  não for acessível a partir de  $S$ , o passo 5 não pode ser atingido. A parte (b) da demonstração anterior exclui a possibilidade de **ciclar** indefinidamente e portanto o algoritmo **pára**, o

V.1

que só ocorre no passo 1 .

Se o **algoritmo** pára no passo 1,  $T$  não pode ser acessível a partir de  $S$ , pois isso contrariaria o lema (17) .

**19 Teorema** : Suponha-se que  $H$  satisfaz a **hipótese** (16) e suponha - se que em-alguma **iteração** o **algoritmo** entrou no passo 5, satisfa - zendo nesse instante

$$(\forall n \in T) \quad \hat{h}(n) = 0 \quad , \quad (\forall n \in N) \quad \hat{h}(n) \leq h(n)$$

Nesse caso, o caminho gerado pelo **algoritmo** entre  $S$  e  $\bar{n}$  resolve o problema PB (9) .

**Demonstração** : A demonstração segue [9] diretamente, sendo apresen - tada somente para manter completo o tratamento.

Suponha-se , por absurdo, que  $\hat{g}(\bar{n}) = f(\bar{n}) > h(S)$  .

Seja  $P = (n_1, n_2, \dots, n_p)$  um caminho de custo mfnimo entre  $S$  e  $T$ . Mostra-se inicialmente:

(a) No instante em que  $\bar{n}$  é escolhido, há na lista **Aberto** um nó  $\hat{n} \in P$  com  $\hat{g}(\hat{n}) \leq g(\hat{n})$  (ou, equivalentemente,  $\hat{g}(\hat{n}) = g(\hat{n})$ , por de - finição de  $g$ ) :

Se  $n_1$  está aberto , (a) é satisfeito

Senão, seja  $n_k$  o nó de mais alto índice em  $P$  tal que  $n_k$  está fechado e  $\hat{g}(n_k) = g(n_k)$ .  $k < p$  , pois  $n_p \in T$  e nós em  $T$  nunca são fechados.

Como  $n_k$  está fechado,  $n_k$  foi expandido e  $n_{k+1} \in r(n_k) \cap P$  foi gerado com custo  $\hat{g}(n_{k+1}) = \hat{g}(n_k) + c(n_k, n_{k+1}) = g(n_k) + c(n_k, n_{k+1}) = g(n_{k+1})$  , permanecendo após as eliminações em

V.1

1

Aberto (pois  $n_{k+1}$  não estava em Fechado), com  $\hat{g}(n_{k+1}) \leq g(n_{k+1})$ .

Por hipótese,  $n_{k+1}$  não é fechado posteriormente e (a) é válido com  $\bar{n} = n_{k+1}$ .

Pode-se agora estabelecer a contradição:

No instante em que  $\bar{n}$  é fechado, tem-se

$$\begin{aligned}
 f(\bar{n}) &= \hat{g}(\bar{n}) + 0 && \text{por hipótese} \\
 &h(\bar{n}) && \text{por hipótese de absurdo} \\
 &= g(\bar{n}) + h(\bar{n}) && \text{pois } \bar{n} \text{ pertence a um caminho ótimo} \\
 &&& \text{entre } S \text{ e } T \\
 &\geq \hat{g}(\bar{n}) + \hat{h}(\bar{n}) && \text{usando (a)} \\
 &= \hat{f}(\bar{n})
 \end{aligned}$$

Portanto,  $f(\bar{n}) > \hat{f}(\bar{n})$  o que é absurdo, pois o algoritmo escolheria  $\bar{n}$  no passo 1.

O teorema (19) demonstra a admissibilidade do algoritmo  $\hat{A}$ , juntamente com o lema (17). Os resultados acima podem ser englobados no teorema a seguir, em que o critério de parada é pouco realista.

- 20 **Teorema** : Sejam  $H$  finito sem ciclos de custo negativo,  $T$  acessível a partir de  $S$ . Se  $h$  satisfaz (21) a partir de uma certa iteração, então  $\hat{A}$  com o critério de parada (21) termina com uma solução ótima de PB (9).



21 Critério de parada : O critério é satisfeito se

$$\forall n \in N \quad \hat{h}(n) \leq h(n)$$

$$\forall n \in T \quad \hat{h}(n) = 0$$

Critérios de parada : É difícil elaborar critérios que terminem o algoritmo com a certeza de ter encontrado um caminho de custo mínimo, a não ser que seja conhecida alguma função  $\hat{h}^*$  satisfazendo (21). Nesse caso,  $\hat{h}$  pode ser comparada com  $\hat{h}^*$  ao invés de  $h$ . Tais funções são conhecidas para muitos problemas e encontram-se exemplos relacionados à Inteligência Artificial em [2]. Um estudo relacionado a problemas de decisões sequenciais será feito na seção 3.

É bastante conhecido [3] o fato de que a utilização de funções  $\hat{h}$  que não satisfazem (21) pode levar a soluções rápidas para o problema de busca de caminhos entre  $S$  e  $T$ , mas não se pode garantir sua otimalidade. Este fato é explorado pelo algoritmo  $\hat{A}$ , que pode iniciar com uma heurística "forte" e diminuir  $\delta$ , iterativamente à medida que soluções sub-ótimas são encontradas. Se durante a execução do algoritmo esgotar-se o tempo disponível, pode-se parar, recuperando na lista Aberto a melhor solução encontrada até então.

Uma estratégia de redução de  $\hat{h}$  será sugerida em conexão com problemas de decisão sequencial, onde se mostra também que vários algoritmos de busca em grafos podem ser expressos como casos particulares de  $\hat{A}$ .

## V.1

Um fato interessante sobre  $A$  é demonstrado a seguir, ligado ao problema de encontrar caminhos de custo mínimo a todos os nós de  $T$  acessíveis a partir de  $S$ .

22 **Teorema** : Suponha-se que  $H$  satisfaz à hipótese (16). Suponha-se que em alguma iteração, o passo 1 de  $\hat{A}$  encontrou em Aberto somente nós pertencentes a  $T$ . Nesse caso, qualquer que tenha sido o comportamento de  $\hat{n}$ , tem-se

- (i) Encontram-se em Aberto todos os nós de  $T$  acessíveis a partir de  $S$  por caminhos que não passam por  $T$ .
- (ii) Seja  $n$  um nó de Aberto. Se existe um caminho de custo mínimo entre  $S$  e  $n$  que não passa por  $T$ , então

$$\hat{g}(n) = g(n)$$

**Demonstração** :

- (i) A primeira parte é consequência direta do lema (17) : basta aplicar o algoritmo ao grafo obtido fazendo  $\Gamma(n) = \emptyset \quad \forall n \in T$ , com o alvo  $n$ .

- (ii) Seja  $n$  um nó de Aberto no instante considerado

Seja  $P$  um caminho de custo mínimo entre  $S$  e  $n$ , tal que  $P$  não passa por  $T$ .

Na parte (a) da demonstração do **teorema (19)**, mostra-se que quando  $\hat{A}$  entra no passo 2, existe em Aberto um nó  $\bar{n}$  de  $P$  com  $\hat{g}(\bar{n}) = g(\bar{n})$ .

Por hipótese,  $n$  é o único nó de  $P$  presente em Aberto e, portanto,  $\bar{n} = n$ , o que completa a demonstração.

O **teorema** acima **é** Útil quando se procura encontrar caminhos de custo **mínimo** a todos os nós do alvo, e poderia ser usado para demonstrar a admissibilidade do **algoritmo 111.1.17** .

## Seção 2 - OTIMIZAÇÃO EM GRAFOS DOTADOS DE UMA RELAÇÃO DE PREFERÊNCIA

Em certos problemas de otimização pode existir uma relação de preferência definida no conjunto de nós. Este caso pode ocorrer quando os nós são conjuntos, naturalmente ordenados pela relação de continência, ou quando os nós são vetores de um espaço euclidiano com a ordenação parcial usual (componente a componente).

Em problemas de planejamento é comum aceitar-se que uma configuração para um sistema é superior a outra sempre que a primeira possuir um excesso de instalações sobre a segunda. Este critério de preferência não é geral, mas é frequentemente aplicável, como se comentará no próximo capítulo.

Nesta seção introduz-se o problema de otimização em grafos dotados de uma relação de preferência, examinando as simplificações que pode sofrer o algoritmo  $\hat{A}$  (13) devido a essa relação.

A relação de preferência será definida a partir de uma função utilidade com valores em algum espaço euclidiano.

Considere-se o problema PB (9) .

**23 Definição :** O problema PB é dito parcialmente ordenado se existir uma função utilidade  $u : N \rightarrow E$  onde  $E$  é um espaço euclidiano tal que:

$$(i) \quad ( \forall n \in N, \forall n' \in \Gamma(n) ) \quad \begin{cases} u(n') \geq u(n) \\ u(n') = u(n) \quad c(n, n') = 0 \end{cases}$$

$$(ii) \quad ( \forall n, n' \in N ) \quad u(n') \geq u(n) \iff h(n') \leq h(n)$$

Onde  $\geq$  é a ordenação usual em espaços euclidianos.

v.2

24 **Definição** : A relação de preferência  $\succ$  é definida a partir de  $u$ :

$$(\forall n, n' \in N) \quad n \succeq n' \iff u(n) \geq u(n')$$

Adota-se a notação seguinte:

$$n \preceq n' \iff n' \succeq n$$

$$n \succ n' \iff n \succeq n' \quad \text{e} \quad u(n) \neq u(n')$$

Portanto, se PB é parcialmente ordenado, pode-se escrever (23) no formato mais conveniente:

- 25 (i)  $(\forall n \in N, \forall n' \in \Gamma(n)) \quad n' \succeq n$   
 $n' \approx n \implies c(n, n') = 0$   
 (ii)  $(\forall n, n' \in N) \quad n' \succeq n \implies h(n') \leq h(n)$

Os lemas seguir mostram a estrutura imposta ao grafo  $H$ .

26 **Lema** : Seja PB parcialmente ordenado,  $N$  finito. Nesse caso, qualquer circuito em  $H$  tem custo nulo.

Demonstração ;

Seja  $P = (n_0, n_1, \dots, n_p, n_0)$  um circuito em  $H$ .

Tem-se  $n_0 \succeq n_p \succeq \dots \succeq n_1 \succeq n_0$  devido a (25i) e portanto,

$$n_0 \approx n_1 \approx \dots \approx n_p$$

Usando (25i)

$c(n_{i-1}, n_i) = 0 \quad i=1, 2, \dots, p$ , o que completa a demonstração.

27 **Lema** : Seja PB parcialmente ordenado,  $N$  finito

Seja  $\bar{N} = \{n \in N \mid T \text{ é acessível a partir de } n\}$

V.2

Seja  $H_T = (N_T, \Gamma)$  o subgrafo gerado por  $\bar{N}$ , isto é,

$$N_T = \bigcup_{k=0,1,\dots} \Gamma^k(\bar{N})$$

Nesse caso,

(i)  $(\forall n \in N) \quad h(n) \text{ é finito} \iff n \in N_T$

(ii) Os ramos de  $H_T$  tem custos não negativos

Demonstração:

(i) Seja  $n \in N_T$ . Por definição de  $N_T$ , existe  $k \geq 0$  tal que

$$n \in \Gamma^k(\bar{n}), \quad \bar{n} \in \bar{N}$$

Como  $T$  é acessível a partir de  $\bar{N}$ ,  $h(\bar{n}) < +\infty$

Devido aos lemas (6) e (26),  $h(n) > -\infty$

Portanto  $h(n)$  é finito e devido a (25)

$$h(\bar{n}) \leq h(n) \quad \text{e portanto}$$

$$-\infty < h(n) \leq h(\bar{n}) < +\infty$$

A implicação inversa é trivial, por definição de  $h$  (7).

(ii) Seja  $(n, n')$  um ramo de  $H_T$

Devido a (i) e (25), tem-se

$$-\infty < h(n') \leq h(n) < +\infty$$

Por definição de  $h$ ,

$$h(n) \leq c(n, n') + h(n')$$

Somando as desigualdades, obtém-se  $c(n, n') \geq 0$  e termina a demonstração.

O lema (26) mostra que a hipótese (16) é redundante para problemas com ordenação parcial. O lema (27) mostra que custos negativos podem ocorrer somente em ramos que não podem fazer parte de um caminho terminando no alvo T. Os nós ligados a tais ramos podem ser eliminados de N sem prejuízo, o que pode ser feito por  $\hat{A}$ : eliminam-se nós ligados a ramos de custo negativo. Esta simplificação não será feita, para evitar a complicação de  $\hat{A}$ .

Como se verá no capítulo seguinte, caminhos de custo nulo são importantes em planejamento a longo prazo, mas circuitos devem ser detetados pelo algoritmo. Para evitar a complicação de  $\hat{A}$ , faz-se a hipótese de não existência de circuitos em (28). Observe-se que devido a (26), exclui-se somente o caso "patológico" de circuitos de custo nulo.

28 Hipótese : PB é parcialmente ordenado (23), N é finito e H não tem circuitos.

Se PB satisfizer (28), algumas simplificações poderão ser feitas no algoritmo  $\hat{A}$ , desenvolvidas a seguir com base em um novo enunciado da regra de eliminação (14).

29 Regra de eliminação

$$n \text{ elimina } n' \text{ se e só se } \begin{cases} n \geq n' , & \hat{g}(n) \leq \hat{g}(n') \\ n \text{ não pertence ao caminho encontrado por } \hat{A} \text{ entre } S \text{ e } n' \end{cases}$$

Em outras palavras, elimina-se um nó quando houver outro "melhor" e "mais barato". A segunda condição serve somente para evitar que um nó elimine seus sucessores em um caminho de custo

V.2

nulo. Essa condição é facilmente implementável.

Passa-se à demonstração da admissibilidade do algoritmo  $\hat{A}$  para problemas parcialmente ordenados. Supõe-se nesta seção que o algoritmo usa a regra de eliminação (29).

30 Lema : Suponha-se que PB satisfaz (28) e T é acessível a partir de S. Nesse caso, qualquer que seja a atuação do passo 4, sempre que  $\hat{A}$  entra no passo 1, existe na lista Aberto um nó n satisfazendo

$$31 \quad \hat{g}(n) + h(n) = h(S)$$

Em outras palavras, a qualquer momento foi encontrado um segmento de caminho Ótimo cujo nó terminal está aberto.

Demonstração : Considere-se  $\hat{A}$  entrando no passo 1 em uma iteração qualquer.

Seja  $\bar{N} \subset N$  o conjunto de todos os nós n (inclusive de S) gerados em qualquer iteração até o instante considerado, tais que n satisfaz (31).

Tem-se:  $\bar{N} \neq \emptyset$ , pois algum nó de S satisfaz (31)

Seja  $\tilde{N} \subset \bar{N}$  o conjunto de nós maximais de  $\bar{N}$

(isto é,  $n' \in \tilde{N}$ ,  $n'' \in \bar{N} \rightarrow n'' \not\geq n'$ )

32 Tem-se: um elemento  $\tilde{n}$  de  $\tilde{N}$  somente pode ser eliminado por outro elemento n de  $\bar{N}$  satisfazendo  $n \geq \tilde{n}$ ,  $\hat{g}(n) = \hat{g}(\tilde{n})$ ,  $h(n) = h(\tilde{n})$ .

De fato, seja  $\tilde{n} \in \tilde{N}$

n elimina  $\tilde{n} \Rightarrow n \geq \tilde{n}$  devido a (29)

$$h(n) \leq h(\tilde{n}) \text{ devido a (25ii)}$$

$$\hat{g}(n) \leq \hat{g}(\tilde{n}) \text{ devido a (29)}$$



V.2

somando,  $\hat{g}(n) + h(n) \leq \hat{g}(\tilde{n}) + h(\tilde{n}) = h(S)$

A igualdade vale porque  $h(S)$  é mínimo. Isso e  $n \geq \tilde{n}$  mostram que  $n \in \tilde{N}$ .

Além disso, como  $\tilde{n}$  é maximal,  $n \not\geq \tilde{n}$  e portanto  $n \approx \tilde{n}$ ,  
 $h(n) = h(6), \hat{g}(n) = \hat{g}(6)$

Um elemento de  $\tilde{N}$  somente pode ser eliminado ao ser gerado:

De fato, se  $n$  elimina  $\tilde{n}$ , então  $\tilde{n}$  elimina  $n$ , pois  $n \approx \tilde{n}$ ,  
 $\hat{g}(n) = \hat{g}(\tilde{n})$ . O passo 3 de  $\hat{A}$  inicia pela eliminação dos **nós**  
 gerados por ele.

Se **nenhum** elemento de  $\tilde{N}$  foi fechado, nada **há** portanto a demonstrar,  
 uma vez que  $\tilde{N} \neq \emptyset$  e o primeiro **nó** de  $N$  gerado não foi eli-  
 minado.

Suponha-se portanto que  $\tilde{n}$  foi o Último **nó** de  $\tilde{N}$  a ser fechddo.

Existe um **nó**  $\hat{n} \in \Gamma(\tilde{n})$  tal que  $\hat{n} \in \tilde{N}$  que é comparado com os **nós**  
 listados.

Como  $\tilde{n}$  é maximal,  $\hat{n} \not\geq \tilde{n}$

33 por (25)  $\hat{n} \geq \tilde{n}$  e portanto  $\tilde{n} \approx n$

Consequentemente,  $h(6) = h(\hat{n}), \hat{g}(\tilde{n}) = \hat{g}(6)$

Se  $\hat{n}$  entra em Aberto, termina a demonstração, pois  $\tilde{n}$  foi o Últi-  
 mo a ser fechado e um **nó** de  $\tilde{N}$  **listado** não pode ser eliminado.

Suponha-se por absurdo que  $\hat{n}$  é eliminado por um **nó**  $n$ .

Tem-se  $n \approx \hat{n} \approx \tilde{n}$  devido a (32) e (33)

$\hat{g}(n) = \hat{g}(\hat{n}) = \hat{g}(\tilde{n})$

$\tilde{n}$  não pertence ao caminho entre  $S$  e  $n$ , pois  $\tilde{n}$  estava aberto

$n$  não pertence ao caminho entre  $S$  e  $\tilde{n}$ , pois  $n$  elimina  $\hat{n} \in \Gamma(\tilde{n})$

logo,  $n$  elimina  $\tilde{n}$ ,  $\tilde{n}$  elimina  $n$  e ambos coexistem entre os

nós listados: isso é absurdo, pois o último entre os dois a ser gerado seria eliminado pela regra (29).

O lema a seguir corresponde ao lema (17):

34 Lema: Suponha-se que PB satisfaz (28) e T é acessível a partir de S. Nesse caso, qualquer que seja a atuação do passo 4, o algoritmo  $\hat{A}$  com a regra de eliminação (29) atinge o passo 5 em um número finito de iterações.

Demonstração: Segue-se o método usado na demonstração de (17). O lema (31) garante que o algoritmo não para no passo 1, pois Aberto nunca está vazia. A parte (b) da demonstração continua válida e não precisa ser refeita.

O teorema a seguir corresponde ao teorema (19):

35 Teorema : Suponha-se que PB satisfaz (28). Suponha-se que o algoritmo  $\hat{A}$  com a regra (29) entra no passo 5 em alguma iteração, satisfazendo nesse instante

$$(\forall n \in N) \quad \hat{h}(n) \leq h(n), \quad (\forall n \in T) \quad \hat{h}(n) = 0$$

Nesse caso, o caminho gerado entre S e o nó escolhido  $\bar{n}$  resolve PB.

Demonstração: Devido ao lema (31), quando  $\bar{n}$  é escolhido, existe  $\hat{n}$  em Aberto tal que

36  $\hat{g}(\hat{n}) + h(\hat{n}) = h(S)$

Tem-se:

$$\begin{aligned} \hat{g}(\bar{n}) &= \hat{g}(\hat{n}) + \hat{h}(\bar{n}) && \text{por hipótese, pois } \bar{n} \in T \text{ e } \hat{h}(\bar{n})=0 \\ &= \hat{f}(\bar{n}) && \text{por definição (passo 1 de } \hat{A}) \end{aligned}$$

V.2

$$\begin{aligned}
 &\leq \hat{f}(\hat{n}) && \text{pois } \hat{f}(\hat{n}) \text{ é mínimo, por construção} \\
 &\leq \hat{g}(\hat{n}) + h(\hat{n}) && \text{pois } h(\hat{n}) \geq \hat{h}(\hat{n}) \text{ por hipótese} \\
 &\leq h(s) && \text{devido a (36)}
 \end{aligned}$$

Mas por definição de  $g$ ,  $\hat{g}(\hat{n}) \geq h(s)$  e portanto

$$\hat{g}(\hat{n}) = g(\hat{n}) = h(s) \quad , \text{ o que completa a demonstração .}$$

37 Corolário : Se PB for parcialmente ordenado, são válidos os resultados da seção anterior (17-20) para o algoritmo  $\hat{A}$  com a regra de eliminação (24) .

Demonstração: Os resultados de (17) e (19) foram demonstrados em (34), (35). Os resultados de (18) e (20) seguem imediatamente dos anteriores.

Os resultados acima permitem uma redução dos requisitos computacionais necessários à resolução de problemas em que se define uma ordenação parcial. Este é o caso de problemas de otimização da expansão a longo prazo de sistemas. Nesses problemas a ordem em que os equipamentos são implementados é importante no cálculo dos custos, o que pode dar origem a pares de configurações comparáveis por (29).

Deve-se observar que as eliminações feitas pelo algoritmo tem como finalidade reduzir tempo e memória em sua execução . Desde que o grafo  $H$  não possua ciclos de custo nulo, todas as conclusões acima são válidas para o algoritmo  $\hat{A}$  sem eliminações. Isso conduz à seguinte observação:

38 Observação : A utilização das regras de eliminação em  $\hat{A}$  pode ser

## V.2

suprimida, sempre que isso levar a uma economia em tempo de computação ou memória.

A manipulação das listas Aberto e Fechado pode ser trabalhosa, quando a verificação da regra de eliminação for complexa. O algoritmo  $\hat{A}$  não elimina nós fechados por esta razão, embora possa ser facilmente modificado com esta finalidade, o que é feito em [34]: se o passo 3 gera um nó já presente em Fechado com custo superior, sua expansão é novamente executada por  $\hat{A}$  e o nó existente não é retirado de Fechado. Em [34] não se refaz a expansão, aproveitando-se os resultados já listados. \*

A eliminação total da manipulação da lista Fechado é conveniente quando se puder concluir que poucas eliminações serão feitas por seus elementos. Pode-se nesse caso manter a lista fora da memória rápida do computador, ganhando em tempo a memória.

Em problemas parcialmente ordenados pode-se fazer essa simplificação sem grandes perdas, como se demonstrará no restante desta seção. Os resultados obtidos são válidos somente para o algoritmo  $\hat{A}$  com  $\hat{h}=0$ , isto é, sem utilização de heurística.

39

Suponha-se que  $PB$  é parcialmente ordenado,  $N$  é finito e  $T$  é acessível a partir de qualquer nó de  $S$  (observe-se que nesse caso qualquer ramo de  $H$  tem custo não negativo e  $H$  não tem ciclos (27)). Suponha-se ainda que  $\hat{h}=0$  em qualquer iteração do algoritmo  $\hat{A}$  e que o passo 5 simplesmente termina o algoritmo, que utiliza a regra de eliminação (29).

---

\* Pode-se demonstrar que se a heurística  $\hat{h}$  for constante e satisfizer propriedades semelhantes a uma distância ( $h(n, n') \geq \hat{h}(n') - \hat{h}(n) \forall n, n' \in N$ ) é impossível haver eliminações em Fechado. Ver [6]. Essa propriedade é satisfeita por  $\hat{h}=0$ , o que será estudado adiante.

Com as hipóteses (39), tem-se uma forma modificada do algoritmo de Dijkstra, fazendo uso da ordenação parcial de PB. São evidentemente válidas todas as conclusões sobre a admissibilidade de  $\hat{A}$  obtidas nesta seção.

- 40 **Teorema** : Considere-se o problema PB com as hipóteses (39). Suponha-se que em uma iteração qualquer é gerado no passo 3 um nó  $\bar{n}$  e seja  $\hat{n}$  um nó qualquer fechado. Nesse caso, nenhum entre os dois nós pode eliminar o outro pela regra (29).

**Demonstração** :

Observe-se inicialmente que com  $\hat{n}=0$ , tem-se  $\hat{f} = \hat{g}$ .

Devido ao lema (27), qualquer ramo tem custo não negativo, e portanto

$$n \in N, n' \in \Gamma(n) \implies c(n, n') \geq 0, \text{ ou seja}$$

$$\hat{g}(n') \geq \hat{g}(n)$$

A cada iteração, o algoritmo escolhe o menor valor de  $\hat{f}$  listado e introduz na lista novos valores não inferiores àquele. Portanto, se  $n'$  é fechado depois de  $n$ ,  $\hat{g}(n') \geq \hat{g}(n)$

(a)  $\bar{n}$  não elimina  $\hat{n}$ .

Suponha-se que, por absurdo,  $\bar{n} \geq \hat{n}$ ,  $\hat{g}(\bar{n}) \leq \hat{g}(\hat{n})$

Seja  $P = (n_0, n_1, \dots, n_P, \bar{n})$  o caminho entre  $S$  e  $\bar{n}$  encontrado pelo algoritmo.

No instante em que  $n$  foi fechado, algum nó  $n_k$  de  $P$  estava aberto, pois  $\hat{n}$  foi fechado antes de  $n_P$ .

Tem-se

$$\begin{aligned} \hat{g}(\hat{n}) &\leq \hat{g}(n_k) && \text{pois } \hat{n} \text{ foi fechado antes de } n_k \\ &\leq \hat{g}(n_{k+1}) && \text{pois os custos são não negativos} \end{aligned}$$

V.2

$$\begin{aligned}
&\leq \hat{g}(n_p) \\
&\leq \hat{g}(\bar{n}) \\
&\leq \hat{g}(\hat{n}) \quad \text{por hipótese}
\end{aligned}$$

Portanto,  $\hat{g}(n_k) = \hat{g}(n_{k+1}) = \dots = \hat{g}(\bar{n}) = \hat{g}(\hat{n})$

Devido a (25i) e à hipótese de absurdo,

$$n_k = n_{k+1} \approx \dots \approx \bar{n} \succeq \bar{n}$$

Portanto,  $n_k \succeq \bar{n}$ ,  $\hat{g}(n_k) \leq \hat{g}(\bar{n})$  e  $n_k$  não está no caminho de  $S$  a  $\bar{n}$  (pois ambos coexistem em Aberto).

Finalmente, em alguma iteração coexistiram em Aberto  $n_k$  e  $\bar{n}$ , sendo que  $n_k$  elimina  $\bar{n}$ . Isso é absurdo, pois  $\bar{n}$  seria eliminado na iteração em que o Último entre os dois foi gerado.

(b)  $\hat{n}$  não elimina  $\bar{n}$

Suponha por absurdo que

$\hat{n} \succ \bar{n}$ ,  $\hat{g}(\hat{n}) \leq \hat{g}(\bar{n})$ ,  $\hat{n}$  não está no caminho  $P$  gerado entre  $S$  e  $\bar{n}$

$$P = (n_0, n_1, \dots, n_p, \bar{n})$$

No instante em que  $\bar{n}$  foi fechado, um  $n_k$  de  $P$  estava aberto (como em (a)).

Tem-se, como anteriormente,

$$\hat{g}(\hat{n}) \leq \hat{g}(n_k) \quad \text{pois } \hat{n} \text{ é fechado antes de } n_k$$

$$n_k \leq \dots \leq n_p \leq \bar{n} \leq \hat{n}$$

Portanto,  $\hat{n} \succeq n_k$ ,  $\hat{g}(\hat{n}) \leq \hat{g}(n_k)$ ,  $\hat{n}$  não está no caminho entre  $S$  e  $n_k$ .

## V.2

Logo,  $n$  elimina  $n_k$  e coexiste com  $n_k$  na lista Aberto em alguma iteração, o que é impossível e termina a demonstração.

O resultado acima justifica o abandono completo de comparações com nós fechados quando não se usar heurística. Existem razões para se supor que se a heurística usada for razoavelmente realista, serão raras as eliminações devido a nós fechados e pode-se economizar tempo e memória mantendo a lista Fechado em disco, sem efetuar eliminações. Essa abordagem foi utilizada com êxito na implementação comentada no Capítulo VII .

### Seção 3 - GRAFOS EM CAMADAS E PROBLEMAS DE DECISÃO SEQUENCIAL

Problemas de decisão sequencial em processos determinísticos a horizonte finito admitem uma representação por grafos com uma estrutura especial, a que se chamarã Grafos em Camadas. A aplicação a esses grafos das técnicas desenvolvidas neste capítulo conduz a métodos eficientes para sua resolução.

Seja  $H = (N, \Gamma)$  um grafo, onde  $\Gamma : N \rightarrow P(N)$  é tal que  $(\forall n \in N) \quad \Gamma(n)$  é finito.

41 Definição : O grafo  $H$  é um grafo em camadas se existir uma partição de  $N$

$\{N^i\}_{i \in I}$ , onde  $I = \{0, 1, 2, \dots\} \subset \mathbb{N}$ , tal que

(i)  $N^0$  é finito

(ii)  $(\forall n \in N^i) \quad \Gamma(n) \subset N^{i+1}$  ou  $\Gamma(n) = \emptyset$ ,  $i=0, 1, 2, \dots$

42 As seguintes propriedades seguem imediatamente da definição acima:

Cada camada  $N^i$  de  $N$  é um conjunto finito.

Se  $I$  for finito,  $H$  é finito.

$H$  não tem ciclos.

Define-se a seguir um problema de busca de caminhos de custo mínimo em grafos em camadas (PBc). Trata-se do problema PB(9), em que  $S \subset N^0$  e o alvo  $T$  está contido em alguma camada  $N^f$ . Suponha-se conhecida a função custo  $c$ , definida em (3), e  $h$  definida em (7).



## V.3

- 43 PBC :Seja  $H = (N, T)$  um grafo em camadas (41). Seja  $f \in I$  e considerem-se  $S \subset N^0$ ,  $T \subset N^f$ . Encontrar, se existirem, dois nós  $s \in S$  e  $t \in T$  e um caminho  $G$  unindo  $s$  a  $t$ , tais que  $c(G) = h(S)$ .

A seguir, estabelece-se um enunciado bastante geral pa  
ra problemas de decisões sequenciais a estado finito e horizonte fi  
nito, identificados posteriormente ao problema PBC .

- 44 Sejam  $I = (0, 1, 2, \dots, f)$  uma família de estágios, a  
um espaço de estados finito e sejam  $a_i \subset \sigma$ ,  $i=0, 1, \dots, f-1$  con  
juntos de estados admissíveis,  $U_i$ ,  $i=0, 1, f-1$  conjuntos de  
controles admissíveis,  $U_i \subset U$ , onde  $U$  é o espaço de controles,  
finito.

Considerem-se conhecidos os mapeamentos

- 45  $f_i : \sigma_i \times U_i \rightarrow \sigma$ ,  $i=0, 1, \dots, f-1$

que associam a um estado  $x_i$  e um controle  $u_i$  admissíveis em um  
estágio  $i \in I$ , um estado de a

$$x^i \in \sigma_i, u^i \in U_i \mapsto x^{i+1} = f_i(x^i, u^i) \in \sigma$$

Suponham-se conhecidas funções custo

- 46  $c_i : \sigma_i \times U_i \rightarrow R$ ,  $i=0, 1, \dots, f-1$  tais que  
 $x^i \in \sigma_i, u^i \in U_i \mapsto c_i(x^i, u^i) \in R$

Sejam também conhecidos os mapeamentos

- 47  $\delta_i : \sigma_i \rightarrow P(\sigma_{i+1})$ ,  $i=0, 1, \dots, f-1$

## V.3

que associam a um estado admissível os estados sucessores permitidos no estágio seguinte. \*

- 48 **Definição** : Dada uma sequência  $u=(u^i)_{i=0,1,\dots,f-1}$ , diz-se que a sequência é **admissível** para o estado inicial  $x^0 \in o_0$  se  $u^i \in U_i$ ,  $i=0,1,\dots,f-1$  e existe uma sequência  $x=(x^i)_{i \in I}$  tal que, para  $i=0,1,\dots,f-1$ ,

$$\begin{aligned} x^{i+1} &\in \delta_i(x^i) \\ x^{i+1} &= f_i(x^i, u^i) \end{aligned}$$

Nesse caso, diz-se que a sequência  $u$  gera a trajetória admissível  $x$  e define-se o **custo** da sequência  $u$  por

$$49 \quad \sum_{i=0}^{f-1} c_i(x^i, u^i)$$

- 50 **Problema de Controle Ótimo** : Considera-se neste trabalho o seguinte enunciado para o Problema de Controle Ótimo : Dados  $S \subset \sigma_0$ ,  $T \subset \sigma_f$ , encontrar um estado inicial  $x^0 \in S$ , uma sequência de **controles**  $\hat{u}$  **admissível** para o estado inicial  $x^0$  e a trajetória  $\hat{x}$  gerada por  $\hat{u}$  e  $x^0$ , tais que :

- (i) o estado terminal  $\hat{x}$  satisfaz  $\hat{x}^f \in T$
- (ii) a sequência  $u$  tem custo **mínimo** entre todas as sequências **admissíveis** para estados iniciais em  $S$ , cujas trajetórias tem estados finais em  $T$ .

$\hat{u}$  e  $\hat{x}$  são chamadas, respectivamente, sequência **ótima** de controles e trajetória Ótima.

---

\* O mesmo tipo de restrição pode ser feito para os controles, limitando-os a conjuntos **admissíveis** dependentes do estado. É possível fazer a equivalência entre os dois tipos de restrição.

V.3

Passa-se em seguida à reformulação do Problema de Controle ótimo acima, de modo a colocá-lo no formato PBC (43).

51 Definição : O Grafo Estado-Estágio associado ao Problema de Contro-le ótimo é um grafo  $H = (N, \Gamma)$ , onde

$$N = \{(x^i, i) \mid x^i \in c_i, \quad i \in I\} \quad \text{e}$$

$$(\forall (x^i, i) \in N \quad \text{tais que } i < f)$$

$$\Gamma(x^i, i) = \{(x^{i+1}, i+1) \mid x^{i+1} \in \delta_i(x^i) \cap f_i(x^i, u_i)\}$$

$$(\forall x^f \in \sigma^f)$$

$$\Gamma(x^f, f) = \emptyset$$

O grafo estado-estágio é um grafo em camadas, cujas camadas são

$N^i = \{(x^i, i) \mid x^i \in \sigma_i\}$ . As camadas são finitas, devido à finitude de  $\sigma$ .

Definem-se a seguir os custos associados aos ramos de  $H$  :

52 Definição : Seja  $r = ((x^i, i), x^{i+1}, i+1))$  um ramo de  $H$ . O custo associado a  $r$  é definido por

$$c(r) = \min\{c_i(x^i, u) \mid x^{i+1} = f_i(x^i, u), u \in U_i\}$$

53 observação : Permite-se alguma ambiguidade na notação dos custos, preferindo-se conservar a letra 'c' e diferenciando-se sua aplicação pelos argumentos. Com isso em mente, usar-se-ão para os custos definidos em (52) as notações abaixo, segundo se queira ou não exprimir a dependência dos custos com respeito à camada do

V.3

grafo (estágio  $i \rightarrow i+1$  do processo) .

$$c(r) \equiv c((x^i, i), (x^{i+1}, i+1)) \equiv c(x^i, x^{i+1}, i+1)$$

- 54 **Teorema** : O Problema de Controle ótimo (50) é equivalente ao problema PBC (43) aplicado ao grafo estado-estágio definido em (51).

#### Demonstração

Basta observar que a determinação de uma trajetória Ótima  $\hat{x}$  para o Problema de Controle Ótimo resolve completamente o Problema: uma sequência Ótima de controles é obtida escolhendo  $\hat{u}$  tal que

$$\begin{aligned} 55 \quad c_i(\hat{x}^i, \hat{u}^i) &= \min\{c_i(\hat{x}^i, u^i) \mid \hat{x}^{i+1} = f_i(\hat{x}^i, u^i), u^i \in U^i\} \\ &= c(x^i, x^{i+1}, i+1) \end{aligned}$$

Além disso, qualquer solução ótima de (50) satisfaz (55), obviamente.

Por construção,  $((x^i, i))_{i=0,1,\dots,f}$  é um caminho em  $H$  se e somente se  $(x^i)_{i=0,1,\dots,f}$  é uma trajetória admissível para (46). Portanto, a qualquer solução Ótima de (50), corresponde um caminho de custo mínimo de  $H$ ; a qualquer caminho ótimo em  $H$  corresponde uma solução Ótima de (50), devido a (55).

Os fatos acima são bastante conhecidos, embora não pareça existir nenhum tratamento unificado de problemas de grafos e controle Ótimo. Qualquer algoritmo de busca de caminhos mínimos em grafos pode ser usado na resolução de PBC e comenta-se abaixo a utilização do algoritmo A .

## V.3

O estudo da aplicação de  $\hat{A}$  à resolução de PBC resume-se ao estudo da manipulação da heurística  $\hat{h}$ . Já se observou que o algoritmo  $A^*$  é reproduzido se  $\hat{h}$  depender do estágio do algoritmo e que  $\hat{h}=0$  leva ao algoritmo de Dijkstra. Em grafos em camadas, pode-se associar um valor de  $\hat{h}$  a cada camada, obtendo resultados interessantes, comentados abaixo.

Seja  $A$  um número muito grande, superior ao custo de um caminho qualquer entre  $S$  e  $T$  no problema PBC. Examinam-se expressões para  $\hat{h}$ , dependentes apenas das camadas de  $H$ :

$$56 \quad \hat{h}(n_i) = i A, \text{ para todo } n_i \in N^i, i=0,1,\dots,f$$

Com heurística (56),  $\hat{A}$  reproduz o algoritmo de Programação Dinâmica (da frente para trás): de fato, executa-se a expansão de nós de uma camada somente quando forem esgotados os nós da camada anterior.

$$57 \quad \begin{aligned} \hat{h}(n_i) &= (f-i)\Delta, & \forall n_i \in N^i \text{ tal que } n_i \notin T, i=0,1,\dots,f \\ \hat{h}(n_f) &= (f+1)\Delta, & \forall n_f \in T \end{aligned}$$

, Com a heurística (57),  $\hat{A}$  comporta-se como um algoritmo de busca em profundidade (Ver III.1.17), aplicado a grafos em camadas. De fato, enquanto houver algum nó na camada  $N^i$ , nenhum nó de  $N^{i-1}$  pode ser escolhido pelo passo 1 de  $\hat{A}$ . Fazendo o algoritmo terminar ao entrar no passo 5, tem-se nesse instante em Aberto todos os nós  $n$  de  $T$ , acompanhados dos custos  $\hat{g}(n)=g(n)$ , de acordo com o teorema (22).

Algumas possíveis estratégias para a manipulação de heurística serão comentadas no próximo capítulo, ligadas ao Problema de Planejamento a Longo Prazo.

propõe-se abaixo uma estratégia aplicável em geral ao PBC, cujo desempenho mostrou-se bom quando aplicado a problemas em que  $T$  é acessível a partir de qualquer nó de  $H$  (note-se que este é o caso de problemas de busca parcialmente ordenados):

- 58 Estratégia de manipulação de  $\hat{h}$  : Seja  $A$  um número grande em relação aos custos de caminhos em  $H$ . Deseja-se aplicar  $\hat{A}$  à resolução de PBC (43).

Inicialmente, faça-se

$$\hat{h}(n_i) = (f-i)\Delta \quad \forall n_i \in N^i, \quad i=1,2,\dots,f$$

- 59 Suponha-se que em alguma iteração o algoritmo entra no passo 5 com  $\bar{n}_f \in T$  (isso certamente ocorre, devido a (17)), sendo  $(\bar{n}_0, \bar{n}_1, \dots, \bar{n}_f)$  o caminho encontrado entre  $n_0$  e  $n_f$ .

O passo 4 deverá definir uma nova heurística para cada nível (se não se decidir parar o processo), como a seguir:

Para  $i=1,2,\dots,f-1$ , sejam

$$g_i = \begin{cases} \text{se nenhum elemento de } \Gamma(\bar{n}_{i-1}) \text{ está em Aberto} \\ \min\{\hat{g}(n_i) \mid n_i \in \Gamma(\bar{n}_{i-1}), n_i \text{ aberto}\} & \text{no caso contrário.} \end{cases}$$

Se algum  $\bar{g}_i \neq \hat{g}(\bar{n}_i)$  foi encontrado, define-se

60 
$$\hat{h}(n_i) = \hat{g}(\bar{n}_f) - \bar{g}_i - \delta$$

onde  $\delta > 0$  é um número fixo, pequeno em relação a  $h(S)$ .

Se  $\bar{g}_i = \hat{g}(\bar{n}_i)$  ,  $i=1,2,\dots,f-1$  , define-se

$$61 \quad \hat{h}(n_i) = \hat{h}(\bar{n}_i) - \delta$$

Esta estratégia produz valores decrescentes de  $\hat{h}$ , de modo que em tempo finito será satisfeito o critério (21), o que garante a resolução do problema de busca. A vantagem do processo reside no fato de gerarem-se iterativamente estratégias sub-Ótimas.

O efeito de (60) é o seguinte: após encontrar-se um caminho entre S e T, diminui-se  $\hat{h}$  de modo a forçar o algoritmo a pesquisar novos caminhos em uma "vizinhança" do caminho encontrado. De fato, em cada nível em que (60) foi executado, um nó  $\hat{n}_i$  satisfará  $\bar{g}_i = \hat{g}(\hat{n}_i)$  e

$$\begin{aligned} f(\hat{n}_i) &= \hat{g}(\hat{n}_i) - \bar{g}_i - \delta + \hat{g}(\bar{n}_f) \\ &= \hat{g}(\bar{n}_f) - \delta \\ &= f(\bar{n}_f) - \delta \\ &< f(\bar{n}_f) \end{aligned}$$

e portanto,  $\bar{n}_f$  não poderá ser novamente escolhido pelo passo 1 sem que  $\hat{n}_i$  seja expandido.

Como regra de parada, pode-se limitar o tempo de computação, ou comparar  $\hat{h}$  com alguma função conhecida, satisfazendo (21), ou simplesmente parar o algoritmo quando (60) não puder ser aplicado, o que não garante a otimalidade da estratégia encontrada, mas pode fornecer boas soluções sub-ótimas.

Conclui-se, finalmente, que algoritmos de busca em grafos podem ser usados na resolução de problemas de decisões sequenciais com vantagens sobre o algoritmo de Programação Dinâmica, em problemas com alguma estrutura.

Se os custos forem todos não negativos, o algoritmo de Dijkstra será sempre superior à programação Dinâmica da frente para trás. O algoritmo  $A^*$  com uma heurística satisfazendo (21), se for conhecida alguma, é superior ao algoritmo de Dijkstra, expandindo um número de nós que tende a diminuir quando  $h$  aumenta.

O algoritmo  $\hat{A}$  tem sua importância no fato de gerar rapidamente soluções sub-Ótimas, o que não ocorre com os algoritmos anteriores. Se não se desejarem soluções sub-Ótimas,  $\hat{A}$  perde sua finalidade, podendo-se iniciar a busca com uma heurística satisfazendo (21). No entanto, a obtenção de uma solução Ótima pode ser computacionalmente inviável, para problemas de grande porte, o que exclui a utilização de  $A^*$  ou de Programação Dinâmica. À resolução desses problemas destina-se o algoritmo  $A$ .



## CAPÍTULO VI

### OTIMIZAÇÃO A LONGO PRAZO

Neste capítulo comenta-se a aplicação do **algoritmo** A desenvolvido no capítulo anterior à resolução do Problema de **Otimização** a Longo Prazo para sistemas descritos por redes. Este problema de decisão sequencial pode ser imediatamente colocado sob o formato de problema de **otimização** em grafo em camadas, ao qual aplica-se  $\hat{A}$ . Faz-se na primeira seção a adaptação de PLP, comentando-se em seguida a existência de uma ordenação parcial **no grafo** correspondente a esse-problema.

Particulariza-se em seguida essa aplicação de  $\hat{A}$  para o caso de sistemas de transmissão de energia **elétrica**.

## VI.1

Seção 1 - RESOLUÇÃO DE PLP

Parte-se do enunciado do problema PLP em 11.3.39. Esse problema jã se encontra em um formato muito semelhante ao do problema de busca de caminhos de custo mínimo em grafos em camadas PBC exposto em V.3.43 . É suficiente identificar seus elementos, caracterizando o grafo estado-estágio correspondente a PLP.

Nessa identificação , permite-se alguma ambiguidade na notação, o que não traz confusão e evita carregar excessivamente o formalismo empregado.

1 O grafo estado-estágio  $H=(N_H, \Gamma_H)$  (Ver V. 3.51) correspondente a PLP é o grafo gerado pelo centro  $(c^0, 0)$ , com o operador sucessor

$$2 \quad \Gamma_H(c^i, i) = \{(c^{i+1}, i+1) \mid c^{i+1} \in \Gamma(c^i, (N^i - M^i), P^i, i+1)\} ,$$

onde  $c^i \in N_H$  ,  $i=0, 1, \dots, f-1$

$\Gamma$  é o operador sucessor de  $(\bar{N}, \bar{M})$ , definido em 11.2.17 .

3 Como a topologia é fixa para cada estágio, pode-se alternativamente expressar os nós de  $N_H$  pelos pares  $(s^i, i)$  , onde  $c^i = (N^i, M^i, s^i)$  .

Os custos em  $H$  são definidos como em 11.3.24, por

$$4 \quad c(c^{i-1}, c^i, i) = c(c^{i-1}, c^i, P^i, i) = c_t(c^{i-1}, c^i, i) + c_o(c^i, P^i, i) ,$$

$$\text{para } (c^{i-1}, i-1, c^i, i) \in N_H , i=1, 2, \dots, f$$

O enunciado de PBC fica completo com a especificação do

## VI.1

conjunto de nós iniciais e do conjunto alvo. O conjunto de nós iniciais é composto pelo Único nó  $(N^0, 0)$  e o alvo é dado por

$$T = N_H^f, \text{ camada de índice } f \text{ do grafo } H$$

$T$  corresponde ao conjunto de todas as configurações que podem ser obtidas a partir de  $c^0$  através de  $f$  transições de estado, cada uma delas gerando uma configuração  $c^i$  viável no estágio  $i$  do processo. O enunciado de PBC com esses elementos ficaria:

- 5                    Encontrar um nó  $(C^f, f)$  de  $H$  e um caminho entre  $(c^0, 0)$  e  $(C^f, f)$ , cujo custo é mínimo entre todos os caminhos de  $(c^0, 0)$  a  $N_H^f$  em  $H$ .

A equivalência entre (5) e PLP (11.3.39) é evidente, por construção.

Em problemas de planejamento procura-se retardar ao máximo a implementação de novas instalações, devido ao desconto em seu custo. É comum a hipótese de que o ganho nos custos de operação devido ao adiantamento de uma instalação é inferior à perda provocada por esse adiantamento nos custos de implementação. Nesse caso (que é sempre aceito em sistemas de transmissão de potência), devem-se considerar em  $H$  apenas as sucessoras eficientes de uma configuração, como se definiu em 111.1.9. Portanto, faz-se

$$6 \quad \Gamma_H(c^i, i) = \hat{\Gamma}(c^i, i), \quad \forall (c^i, i) \in N_H, \forall i < f$$

Se não se verificar a hipótese acima, algum procedimento de redução de  $\Gamma_H$  é necessário, uma vez que a instalação de e -

## VI.1

equipamento adicional geralmente não inviabiliza um sistema: isso faria com que os conjuntos  $\Gamma_H$  contivessem muitas sucessoras de custo excessivo.

Em problemas de grande porte, os conjuntos  $\hat{\Gamma}(c^i, i)$  podem ter muitos elementos, o que pode tornar excessivo o porte do grafo  $H$ . Uma regra heurística para a redução do tempo necessário à resolução do problema de otimização consiste em reduzir exponencialmente o número de sucessoras permitido para cada configuração. Essa redução é feita no Problema de Expansão Generalizado (III.1.12) : especifica-se para cada camada  $i=0,1,2,\dots,f-1$  um limite de sucessoras  $k_i$  e faz-se

$$\Gamma_H(c^i, i) = \hat{\Gamma}_{k_i}(c^i, i) \quad , \quad \forall (c^i, i) \in N_H \quad , \quad i < f$$

Essa limitação, aliada à manipulação da heurística pelo algoritmo  $A$ , permite a obtenção de soluções sub-ótimas para problemas de grande porte. A resolução prática de um problema de grande porte envolve algumas tentativas em que se relaxam gradativamente as restrições impostas a  $H$ . Isto será comentado adiante, nesta seção e nas seguintes.

## 8 Ordenação parcial de PLP

O problema de otimização a longo prazo admite normalmente uma ordenação parcial, devida ao fato intuitivo de que uma configuração com mais equipamento instalado do que outra costuma ser superior à segunda. Isto, no entanto, nem sempre é verdade, como se comentará na seção seguinte.

## VI.1

A ordenação utilizada pelo algoritmo  $\hat{A}$  pode originar-se em critérios heurísticos, baseados na prática de planejadores experientes, fornecendo soluções possivelmente não Ótimas. Observe-se, no entanto, que a utilização de tais critérios pode abreviar o processo de busca, conduzindo a soluções melhores do que se poderia obter sem critérios heurísticos.

A demonstração de que um dado critério satisfaz as condições da definição de ordenação parcial pode ser complexa, dependendo do formato dos conjuntos de estados sucessores admissíveis  $\delta(.,.)$ . De fato, esses conjuntos podem ser extremamente restritivos, podendo, por exemplo, existir uma única sequência de configurações que satisfaça essa restrição.

Para ilustrar o raciocínio que leva a uma ordenação parcial, mostra-se a seguir como ordenar parcialmente o problema PLP, baseado em uma hipótese bastante realista sobre sistemas descritos por redes, e utilizando conjuntos de estados sucessores admissíveis bastante simples.

9 Dada uma configuração  $C=(N,M,S)$  e um estágio  $i$ , os conjuntos

$\delta_p(S_p, i)$ ,  $p=1,2,\dots,n$  terão o formato

$$\delta_p(S_p, i) = \{s \in \sigma_p \mid \gamma_p(s) \geq \bar{\gamma}_p\},$$

onde  $\bar{\gamma}_p \geq \gamma_p(S_p)$  depende de  $S_p$  e de  $i$ .

Supõe-se, além disso, que os conjuntos  $\sigma_p$  tem o formato  $\sigma_p = \{s \mid 0 \leq \gamma_p(s) \leq \hat{\gamma}\}$ , onde  $\hat{\gamma}$  é muito grande em relação

## VI.1

às capacidades que possam surgir durante o planejamento.

- 10 Supõe-se ainda que, a cada transição de estado elementar corresponde a implementação de uma instalação, e que a partir de cada estado pode-se fazer qualquer uma entre essas implementações (desde que não seja violada a restrição (9)). O aumento de capacidade e o custo devido a cada implementação independem do estado inicial.

O formato acima para os conjuntos  $\delta(S, i)$  não é sempre excessivamente particular, como se comentará na próxima seção.

A definição a seguir proporcionará critérios para a ordenação de configurações:

- 11 Definição : Sejam  $C = (N, M, S)$  e  $C' = (N', M', S')$  configurações de uma mesma rede básica. Diz-se que  $C$  está contida em  $C'$ ,  $C \subset C'$ , se

$$(i) \quad (N, M) \subset (N', M') \quad \text{segundo II.1.1}$$

$$(ii) \quad \gamma_i(S_i) \leq \gamma_i(S'_i) \quad , \quad i=1, 2, \dots, m$$

Com  $S \in R^m$ ,  $S' \in R^{m'}$ , e com a convenção usual de que incidem os primeiros  $m$  ramos das duas topologias.

Em outras palavras, uma configuração  $C'$  contém  $C$ , quando cada ramo de  $C'$  tem instalações com capacidade maior, ou igual à capacidade das instalações do ramo correspondente em  $C$ . Em muitos sistemas, a viabilidade de  $C$  implica na viabilidade de qualquer configuração de mesma topologia que a contenha: este é normalmente o caso em problemas de transporte, onde não intervem a segunda lei de Kirschhoff,

## VI.1

Quando os testes de viabilidade incluírem situações de emergência, é necessário que a relação (10ii) mantenha-se válida em qualquer dessas situações: o problema surge quando  $C'$  tem uma instalação de maior capacidade substituindo duas ou mais instalações de  $C$ . Em uma situação de emergência neste ramo seria invertida a relação (11ii).

Para cada sistema em particular, deve-se definir uma relação de preferência, se isto for possível. Para o desenvolvimento a seguir, supõe-se que existe uma tal relação, definida a partir da relação de continência (11).

- 12 Hipótese: Supõe-se conhecida uma relação de ordem parcial  $\succeq$  no conjunto de todas as configurações de uma rede básica dada, tal que, dada uma configuração  $C$ , viável no estágio  $i$ ,

$$C' \succeq C \implies C' \text{ é viável no estágio } i.$$

Esta relação deve basear-se na relação de continência (11) de modo a admitir uma função utilidade  $v$  que associa a cada configuração  $C$  um vetor  $v(C)$  tal que

$$v(C') \succeq v(C) \iff C' \succeq C$$

Deve-se ter, além disso,

$$S' \in \delta(S, i) \implies C' \succeq C$$

A função  $v$  deve ser separável, isto é, cada componente de  $v$  deve depender do estado de um ramo de  $C$ . Além disso, a adição de uma mesma instalação a  $C'$  e  $C$  deve manter a relação de ordem.

- 13 Observe-se que se  $\preceq$  coincidir com a relação  $\subset$ , e se os conjuntos de estados sucessores admissíveis forem como se estabe

## VI.1

leceu acima (9), são satisfeitas as condições da hipótese (12), com  $v(C) \in \mathbb{R}^{\bar{m}}$  dada por

$$\begin{aligned} v_i(C) &= \gamma_i(s_i) & \text{se } i < m \\ v_i(C) &= \bar{\gamma}_i(\bar{s}_i) & \text{se } m < i \leq \bar{m} \end{aligned}, \quad (\text{Ver 11.3.32})$$

O caso em que se consideram situações de emergência será estudado na próxima seção ligado a sistemas de transmissão de potência.

O estudo da ordenação parcial do Problema de Planejamento a Longo Prazo fica completo com algumas hipóteses sobre os custos.

#### 14 Hipótese

- (i) Considere-se uma transição elementar de estado  $(s, s')$  do ramo  $r_k$  correspondente a uma certa instalação (10), e sejam  $j > i$  estágios de PLP. Nesse caso, o custo de uma transição de estados  $(\bar{s}, \bar{s}')$  correspondente à mesma instalação no estágio  $j$  satisfaz

$$c_k(\bar{s}, \bar{s}', i) \leq c_k(s, s', i)$$

isto é, o custo de uma instalação é decrescente com o estágio.

- (ii) Dadas duas configurações  $C, C'$  viáveis no estágio  $t$  com relação ao vetor de parâmetros dos nós  $P$ , tem-se

$$C' \succeq C \implies c_0(C', P, t) \leq c_0(C, P, t)$$

- (iii) Nas condições (ii),

$$c_0(C', P, t) - c_0(C, P, t) < c_t(C, C', t),$$

ou seja, a economia no custo das perdas não justifica instalações adicionais.



## VI.1

A primeira hipótese é sempre satisfeita, uma vez que se utilizam custos descontados, o que será exemplificado na próxima seção. A segunda hipótese é menos realista, mas bastante comum. Em sistemas elétricos é normalmente satisfeita e pode em geral ser tomada como uma hipótese heurística: um aumento das instalações existentes diminui os custos de operação, devido à diminuição da carga nas instalações. Note-se que dificilmente um bom planejamento introduz instalações sub-utilizadas, o que reforça a hipótese feita acima. A terceira hipótese é simplesmente uma repetição da hipótese de consistência III.1.11.

Com essas hipóteses, pode-se ordenar parcialmente o problema PBC (5) correspondente ao Problema de Planejamento, o que será feito em dois casos, segundo existam ou não custos de operação:

Sejam  $(c^i, i)$  e  $(c^j, j)$  nós do grafo  $H=(N_H, T_H)$  (1).

15 1º caso : custos de operação positivos:

$$(c^j, j) \geq (c^i, i) \iff c^j \geq c^i \text{ e } j \geq i$$

16 2º caso : custos de operação nulos

$$(c^j, j) \geq (c^i, i) \iff c^j \geq c^i$$

Para demonstrar que PBC é parcialmente ordenado com essas relações, segundo V.2.23, é suficiente mostrar o seguinte lema:

17 Lema: Sejam  $(c^j, j), (c^i, i) \in N_H$ . Nesse caso,

$$(c^j, j) \geq (c^i, i) \implies h(c^j, j) \leq h(c^i, i).$$

O restante da definição da ordenação parcial V.2.23 é

## VI.1

consequência imediata da definição de  $\succeq$  e da hipótese (12).

Demonstração

1º caso: custos de operação positivos

Seja  $\bar{g} = ((\bar{c}^k, k))_{k=i, i+1, \dots, j, \dots, f}$  um caminho de custo mínimo entre  $C$  e  $T$ .

Por definição de  $k$ ,

$$h(\bar{c}^i, i) = c(\bar{g}) \quad , c \text{ definido em (4).}$$

$$18 \quad = c((\bar{c}^k, k))_{k=i, i+1, \dots, j+1} + c((\bar{c}^k, k))_{k=j+1, \dots, f}$$

Constrói-se a seguir um caminho a partir de  $(c^j, j)$ , tal que

$$c^k \succeq \bar{c}^k \quad , \quad k=j, j+1, \dots, f :$$

Seja  $c^{j+1}$  obtido a partir de  $c^j$  adicionando todas as instalações correspondentes à primeira parte do caminho (18) (isto é possível devido às hipóteses sobre  $\delta(s^k, k)$  em (9)).

Devido a (14i) :

$$19 \quad c_t(c^j, c^{j+1}, j+1) \leq c((\bar{c}^k, k))_{k=i, \dots, j+1} \quad ,$$

onde  $c_t$  é o custo de transição de estados.

Para  $k > j+1$ , reproduzem-se as instalações correspondentes à segunda parte do caminho (18), obtendo-se um caminho

$$((c^k, k))_{k=j, j+1, \dots, f} \quad \text{tal que}$$

$$c^k \succeq \bar{c}^k \quad , \quad k=j, j+1, \dots, f \quad , \text{ por construção, uma vez que}$$

$c^j \succeq c^i$  e todas as instalações adicionadas a  $c^i$  foram também adicionadas a  $c^j$ . Por construção, os custos de transição satisfazem

$$20 \quad c_t(c^k, c^{k+1}, k+1) = c_t(\bar{c}^k, \bar{c}^{k+1}, k+1) \quad , \quad k=j+1, \dots, f-1$$

## VI.1

Devido a (14ii), os custos de operação satisfazem

$$21 \quad c_o(c^k, P(k), k) \leq c_o(\bar{c}^k, P(k), k) \quad , \quad k=j+1, \dots, f$$

Somando (19), (20) e (21), obtém-se

$$\begin{aligned} c(\bar{G}) &\leq c(G) \\ &\geq h(G) \quad , \text{ por definição de } h \quad , \end{aligned}$$

o que completa a demonstração, pois  $c(\bar{G}) = h(\bar{G})$

25 caso : custos de operação nulos

Se  $j \geq i$  , recai-se no caso anterior

Se  $j < i$  , tem-se :

Seja  $\bar{G} = ((\bar{c}^k, k))_{k=0,1,\dots,i}$  um caminho qualquer entre  $(c^0, 0)$  e  $(c^i, i)$  .

$$\begin{aligned} c^j &\geq \bar{c}^i \quad \text{por hipótese} \\ &\geq \bar{c}^{i-1} \quad \text{devido a (12)} \\ &\vdots \\ &\geq \bar{c}^j \end{aligned}$$

Portanto,  $c^j$  é viável nos estágios  $j, j+1, \dots, i$  e

$$c(\bar{G}) = 0$$

Portanto, a configuração  $c^j$  permanece inalterada entre os estágios  $i$  e  $j$  , recaindo-se em seguida no caso anterior, terminando a demonstração.

A ordenação parcial apresentada acima exemplifica o método geral. Problemas particulares, com conjuntos de estados sucessores admissíveis e funções utilidade mais complexas do que as usadas acima, podem admitir ordenações de verificação mais difíceis.

## VI.1

Uma vez estabelecidas as idéias gerais sobre a ordenação, bem como sobre as outras simplificações sugeridas para o algoritmo  $\hat{A}$ , deve-se estudar cuidadosamente cada aplicação, procurando métodos para abreviar a busca em cada caso particular. Para problemas pequenos (como o primeiro exemplo do Capítulo VII), pouco cuidado é necessário, podendo-se mesmo usar o algoritmo de Dijkstra. À medida que o porte do problema cresce (como no Último exemplo do próximo capítulo)', torna-se necessário aumentar cada vez mais o esforço de simplificação.

## VI.2

## Seção 2 - OTIMIZAÇÃO A LONGO PRAZO DE REDES DE TRANSMISSÃO DE POTÊNCIA

Como se comentou no início da tese, todo o trabalho exposto procura ser diretamente utilizável à otimização de redes de transmissão de potência. Aconselha-se portanto o uso do algoritmo  $\tilde{A}$  com a estratégia de manipulação da heurística exposta em V.3.58. A representação das configurações encontra-se no Capítulo IV, bem como a descrição do modelo de desempenho utilizado.

Resta somente fazer alguns comentários sobre a ordenação parcial do grafo de configurações  $H$  e sobre o modelo de custos empregado na otimização.

### 22 Modelo de custos

Cada transição elementar de estado corresponde à construção de uma linha de transmissão (ou a uma substituição, o que não ocorre nos sistemas brasileiros e não será comentado).

O custo de construção de uma linha será calculado a partir de um custo por unidade de distância especificado para cada tipo de linha, ou por alguma expressão mais complexa envolvendo terminais e outros fatores.

Esse custo é descontado para o ano inicial, deduzindo-se ainda um valor de revenda da linha calculado considerando uma depreciação linear do equipamento durante um certo tempo.

Seja, portanto,  $\tau$  uma transição de estado correspondente ao estágio  $i$ ; seja  $(t_0, t_1, \dots, t_f)$  a sequência de instantes correspondentes aos estágios de planejamento, com intervalos em

## VI.2

torno de um ano, em geral, Tem-se :

$$23 \quad c_t(\tau, t_i) = c_t^0(\tau) e^{-\beta(t_i - t_0)} - R e^{-\beta \lambda}$$

onde

$c_t^0(\tau)$  é o custo de transição  $r$  calculado a preços de  $t_0$ , sem revenda .

$R$  é um valor residual, dado por

$$R = c_t^0(\tau) \frac{50 + t_i - t_f}{50} \quad \text{para } t_f \leq 50, \text{ instantes em anos .}$$

$\beta$  é uma taxa de desconto e  $\lambda \geq t_f$  é um instante em que se imagi-na revender o sistema ( $\lambda \approx t_f + 10$ ) .

Considerou-se acima que uma linha de transmissão é de-precitada linearmente, caindo seu valor de 2% ao ano. Certamente outros valores podem ser utilizados.

- 24 Custos operacionais : Consideram-se somente os custos das perdas, sendo  $c_0(c^i, P(i), i)$  o custo das perdas térmicas nas linhas en-tre os estágios  $t_i$  e  $t_{i+1}$ ,  $i = 1, 2, \dots, f$  e entre  $t_f$  e  $A$  .

O cálculo das perdas durante um intervalo de tempo é feito por meio de um fator de carga médio do sistema  $F_c$  [24], a par-tir das perdas de ponta  $L$  calculadas em IV.3.34 .

O custo anual descontado será:

$$c_0(C, P, t) = (c_1 L + c_2 F_c^2) e^{-\beta t} ,$$

onde  $c_1$  e  $c_2$  são, respectivamente, o custo anual correspondente à ponta (energia de substituição) e à energia firme,  $F_c \approx 0,55$  .

As hipóteses (14) são consideradas válidas para siste-mas de transmissão, sendo (14ii) de natureza heurística.

## VI.2

25 Ordenação Parcial de H

Comenta-se a seguir o seguinte critério de ordenação parcial, correspondente à relação (12), onde  $C = (N, H, S)$  ,  
 $C' = (N', M', S')$  .

- 26  $C' \succeq C$  se  $(N, M) \subset (N', M')$  e para  $i=1, 2, \dots, m = |M|$   
 ou  $S_i = S'_i$   
 ou  $S'_i$  pode ser obtido a partir de  $S$  construindo linhas no  
 ramo  $r_i$  .

(Como de costume, coincidem os primeiros elementos de  $M$  e  $M'$ )

Com este critério, os conjuntos  $\delta_p(S_p, i)$  podem assumir o formato bastante realista (com a notação em (9)) .

- 27  $\delta_p(S_p, i) = \{s \in \sigma_p \mid \bar{\gamma}_p \leq \gamma_p(s) \leq \hat{\gamma}_p(i)\}$  ,

onde

$\bar{\gamma}_p \geq \gamma_p(S_p)$  depende de  $S_p$  e de  $i$

$\hat{\gamma}_p(i)$  depende do estágio  $i$  e satisfaz

$\hat{\gamma}_p(i) \geq \hat{\gamma}_p(i-1)$  ,  $i=2, 3, \dots, f$  .

São evidentes os seguintes fatos:

Se  $(c^{i+1}, i+1) \in \Gamma_H(c^i, i)$  , então  $c^{i+1} \succ c^i$

Se  $C' \succeq C$  e  $\bar{c}'$  ,  $\bar{c}$  correspondem às configurações obtidas na con  
tingência de um mesmo ramo em  $C'$  e  $C$  , respectivamente, então  
 $\bar{c}' \succeq \bar{c}$  .

Não se pode, no entanto, mostrar que a viabilidade de  $C$   
 em um estágio  $i$  implica na viabilidade no mesmo estágio de qualquer  
 configuração  $C'$  tal que  $C' \succeq C$  . Na realidade, linhas em excesso

## VI.2

podem inviabilizar um sistema, pois o reforço de um ramo pode provocar um fluxo excessivo em outro ramo em série com o primeiro.

A implicação mencionada acima, no entanto, é aceita heurísticamente, devido ao seguinte: uma linha em excesso pode sempre ser desligada do sistema durante a operação, embora isso não seja previsto no modelo.

A partir do critério (26), pode-se portanto implementar a regra de eliminação de  $\hat{A}$  com o critério (15). A demonstração de que o problema é parcialmente ordenado é imediata, e segue diretamente a demonstração do lema (17), a partir dos fatos expostos acima.

## 28 Eficiência de uma configuração

Na otimização a curto prazo, detetou-se a ineficiência 111.1.9, de uma configuração  $C'$  ao constatar-se a existência de outra configuração  $C$  tal que  $C' > C$ . De fato, a correspondência entre transições de estado e construções de linhas faz com que exista apenas um caminho entre  $C^0$  e cada configuração no grafo de expansão (Ver III.1.9). Consequentemente, a definição de ineficiência reduz-se ao critério de ordenação (26).



## CAPÍTULO VII

### IMPLEMENTAÇÃO DO MÉTODO

O método proposto para a resolução do Problema de Planejamento a Longo Prazo de Sistemas de Transmissão de Potência foi implementado <sup>†</sup>, fazendo uso de todas as simplificações comentadas na tese.

Escreveu-se um programa em PL-1 capaz de tratar sistemas de cerca de trinta ramos ocupando 100 Kbytes de memória rápida em um computador IBM 370/145. O programa não impõe nenhuma limitação, em princípio, ao porte do sistema tratado, embora o tempo de computação e a memória ocupada cresçam bastante com o porte do sistema. Já se obtiveram resultados para redes com cinquenta ramos (ver [28]), tendo sido necessária uma grande interação com o programa para obter bons resultados.

Comentam-se neste capítulo alguns aspectos sobre o programa existente, mostrando-se dois exemplos de sua aplicação. Fazem-se finalmente algumas observações sobre procedimentos usados no planejamento de sistemas complexos.

---

<sup>†</sup> A implementação foi feita por uma equipe da COPPETEC, trabalhando em estreito contato com a equipe de planejamento energético da Companhia Furnas - Centrais Elétricas S.A.

## VII.1

Seção 1 - O PROGRAMA

Uma grande economia de memória foi obtida dividindo-se o programa em quatro seções em overlay: **inicialização**, algoritmo A expansão e final, comentadas a seguir.

A topologia foi mantida constante durante todo o planejamento, o que sempre é possível quando a configuração inicial satisfaz a condição de conexidade IV.1.17. Assim, uma configuração é representada por seu estado, guardado na memória sob a forma de uma cadeia binária, criada por uma rotina de compactação.

O programa só admite um tipo de linha por ramo em seu estágio atual, devido às características dos sistemas brasileiros em estudo. Ramos com linhas de diversos tipos devem ser desmembrados em tantos ramos quantos sejam esses tipos.

Inicialização:

Leem-se os dados de entrada, que são colocados no formato usado pelas outras seções do programa, calculando-se capacidades e custos por linha. Entram como dados os parâmetros por quilômetro para cada tipo de linha, a topologia, os comprimentos dos ramos as injeções correspondentes a cada estágio e mais alguns poucos parâmetros. Monta-se a matriz de capacidades, fazendo-se a sua inversão (esta é a única inversão de matrizes em todo o programa, como se comentou em IV.1.32 ).

## VII.1

Expansão:

O algoritmo de expansão corresponde ao algoritmo 111.2.30, com as simplificações comentadas no capítulo III. A lista Fechado é limitado a um comprimento pré-determinado e particionada segundo os níveis. As listas de opções constam simplesmente de índices de ramos, uma vez que a cada ramo associa-se um Único tipo de linha.

Utiliza-se a análise de contingências exposta no capítulo IV e reinversões de matrizes são feitas por meio de IV.1.32. A árvore gerada pelo algoritmo de expansão pode ser listada, e um exemplo consta da próxima seção. As descrições completas das sucessoras obtidas são guardadas em um arquivo em disco.

Algoritmo Â:

Este setor do programa manipula as listas Aberto, Fechado e Lista de sucessores, segundo o algoritmo Â. A lista Fechado permanece em disco, não se fazendo comparações com nós fechados. A lista Aberto consiste de vetores de estado compactados, como se comentou acima, acompanhados de custos. A cada elemento de Aberto, corresponde uma descrição completa guardada em disco, necessária à expansão.

Final:

Esse setor manipula a heurística e é responsável pela saída de dados, que pode listar mais ou menos dados, segundo a quantidade de informação desejada.

O tempo de computação para um problema é consumido quase totalmente pelo setor de expansão, devido à complexidade da análise de contingências. Em seu estágio atual, o programa executa cerca

## VII.1

de cem análises de contingências completas (com a montagem da lista de opções ) por minuto, para um sistema com cerca de trinta nós. Estima-se que esse número possa ser duplicado facilmente aumentando a eficiência do programa. ■

## VII.2

Seção 2 - ESTUDO DE CASOS

Nesta seção, mostram-se resultados da aplicação do método a dois sistemas de transmissão reais: o sistema CEMIG e o sistema FURNAS.

Sistema CEMIG

Trata-se de um sistema modelado por uma rede de pequeno porte, com 6 nós e 9 ramos. Devido ao porte reduzido, o problema de planejamento foi resolvido sem usar heurística, isto é, com  $\hat{h}=0$  no algoritmo  $\hat{A}$  (obtendo-se portanto o algoritmo de Dijkstra modificado).

Uma estratégia ótima foi encontrada em 15 minutos de computação após a realização de 119 expansões, totalizando 479 análises de contingência.

Os dados para o sistema CEMIG encontram-se nas tabelas 1, 2, e 3, onde são listadas as características dos tipos de linhas permitidos (tabela 1) e os dados referentes aos ramos e nós (tabelas 2 e 3). Correspondendo a cada ramo, o programa calcula os incrementos em custo e em capacidade devidos à adição de uma linha padronizada em cada ramo. A cada nó, associam-se as injeções de potência correspondentes a cada estágio.

Após a tabela 5, encontra-se uma cópia de listagem da árvore gerada pelo algoritmo de expansão aplicado à configuração inicial. Cada nó da árvore é representado pelo seu índice na lista de opções de onde foi retirado, e pelo índice do ramo reforçado nessa transição de estado. Segue-se à árvore a lista de Sucessoras obtida pela expansão, ordenada por seus custos (já descontados). Os es-

## VII.2

tados correspondem aos acrêscimos de linhas feitos a partir da configuração inicial, com  $S^0 = 0$ .

A solução Ótima obtida está representada na figura VII.1. Estipulou-se uma taxa de desconto  $\beta = 0,1$ , convencionando-se que o sistema será revendido em 1996. O custo total da estratégia foi 63.69 milhões de dólares, dos quais 54.02 correspondem a custos de implementação.

Embora o sistema estudado seja real, não se procurou usar dados realistas com respeito às condições de viabilidade. A defasagem máxima permitida em cada ramo é de 0,5 radianos, o que é pessimista. O programa calcula limites de ângulo devido ao limite térmico de cada linha, obtendo-se os ângulos de tabela 2. Nota-se imediatamente que há uma discrepância entre os limites de defasagem para os ramos 3 e 7 (somando 0,77 radianos) e o limite para o ramo 4 (0,5 radianos), que está em paralelo com os outros. Esta é possivelmente a razão de não ter sido utilizado o ramo 4, que provavelmente levaria a um projeto de custo mais reduzido, uma vez que a linha padrão para este ramo introduz economias de escala. Esta é uma limitação inerente ao modelo de desempenho adotado e pode ser atenuada através de uma boa "sintonização" dos limites de ângulo impostos aos vários ramos, o que será comentado no capítulo seguinte.

## VII.2

Sistema FURNAS

O sistema Furnas, com 16 nós e 31 ramos já apresenta problemas bem maiores que o sistema CEMIG. Sua otimização foi descrita na referência [11], apresentando-se neste trabalho os esquemas correspondentes à sua otimização. Com exceção dos ramos 20 e 21, que admitem linhas de 800 KV, os parâmetros coincidem com os usados no sistema CEMIG. As tabelas 1, 4 e 5 fornecem os dados usados pelo programa.

A rede básica para o sistema encontra-se na figura IV.1. A estratégia incremental é esquematizada na figura V11.2 e a estratégia ótima (ou sub-Ótima, uma vez que se usou heurística) consta da figura V11.3.

A estratégia incremental constrói 30 linhas, em 16 ramos, ao custo total descontado de 95.3 milhões de dólares, contra 24 linhas em 16 ramos para a política Ótima, cujo custo é 93.2 milhões de dólares. Comparando os esquemas, observa-se que as principais diferenças estão no grupo de pequenos ramos da região noroeste do sistema, onde a segunda política é muito superior à primeira.

Uma observação interessante sobre este sistema está no tratamento das desconexões, que ocorrem em dois casos: em 1979, a contingência do ramo 10 desconecta o nó 4, onde existe geração, o que provoca o reforço de 10 ou 9. Neste mesmo ano, a contingência do ramo 21 desconecta 16 e nenhuma linha é contruída, uma vez que nesse ano a injeção em 16 é nula.

Um exemplo mais complexo encontra-se em [28], correspondente ao sistema interligado CESP-CEMIG-FURNAS, com 50 ramos.

## VII.2

TIPO	TENSÃO KV	RESISTÊNCIA $\Omega/\text{Km}$	REATÂNCIA $\Omega/\text{Km}$	CUSTO $10^3 \$/\text{Km}$	FLUXO MÁXIMO Mw
1	345	0.03400	0.3650	30.5	600
2	500	0.02370	0.3500	45.9	1300
3	800	0.01280	0.3780	88.0	3000

**Tabela 1: Descrição dos tipos de linhas**

RAMO	TIPO	$\bar{\phi}(i)$ Rad	DE	PARA	COMPR. Km	CUSTO $10^6 \$$	$\gamma$ $10^3 \text{ Mw}$	LINHAS INIC.	RAMO
1	2	0.50	0	2	357	16.8	1.99	1	1
2	1	0.19	0	1	103	3.23	3.14	1	2
3	1	0.35	0	6	186	5.84	1.74	2	3
4	2	0.50	0	4	400	19.0	1.78	0	4
5	2	0.48	0	3	264	12.5	2.69	0	5
6	2	0.30	2	3	166	7.90	4.28	1	6
7	1	0.42	4	6	225	7.13	1.44	2	7
8	1	0.32	4	5	170	5.40	1.90	1	8
9	1	0.46	5	6	250	7.96	1.29	0	9

**Tabela 2: Dados para os ramos - sistema CEMIG**  
Custos e capacidades correspondem a uma linha



## VII.2

Nº	1979	1980	1981	1982	1983	1984	1985	1986
1	345	380	425	430	410	420	415	430
2	745	835	920	1175	1335	1375	1350	1415
3	0	0	0	0	0	0	0	0
4	-880	-1010	-1130	-1340	-1380	-1360	-1416	-1545
5	-210	-225	-240	-270	-310	-350	-380	-435
6	-85	-90	-100	-105	-115	-130	-130	-150

**Tabela 3: Dados para os nós Sistema CEMIG**  
**Injeções líquidas em Mw.**

Nº	1979	1980	1981	1982	1984	1986
1	-143	-180	-191	-196	-260	-330
2	1360	1245	1160	1300	1300	1310
3	225	185	159	178	157	135
4	-217	687	1460	1643	1555	1388
5	450	450	450	450	450	450
6	-450	-450	-450	-450	-450	-450
7	1075	985	920	1030	1030	1040
8	245	220	205	235	235	235
9	903	888	839	994	561	270
10	0	0	0	0	0	0
11	-1910	-2360	-2734	-3093	-3985	-5035
12	0	0	515	580	580	585
13	0	0	0	0	0	0
14	-955	-955	-1537	-2181	-3646	-4643
15	-408	-450	-504	-567	-716	-880
16	0	0	0	395	3560	6370

**Tabela 4: Dados para os nós - Sistema FURNAS**  
**Injeções líquidas em Mw**

## VII.2

RAMO	TIPO	COMPRIMENTO Km	CUSTO $10^6$ \$	$\gamma$ $10^3$ Mw	LINHAS INIC.	RAMO
1	1	65	1.81	4.97	1	1
2	1	45	1.25	7.18	1	2
3	1	115	3.20	2.81	1	3
4	1	85	2.37	3.80	1	4
5	1	22	0.61	14.7	1	5
6	1	145	4.03	2.23	1	6
7	1	40	1.11	8.08	1	7
8	1	105	2.92	3.08	1	8
9	2	200	8.40	3.56	0	9
10	1	205	5.70	1.58	1	10
11	2	280	11.8	2.54	0	11
12	2	300	12.6	2.37	0	12
13	1	198	5.51	1.63	2	13
14	2	118	4.96	6.03	2	14
15	2	70	2.94	10.20	1	15
16	2	190	7.98	3.74	1	16
17	2	200	8.40	3.56	1	17
18	1	100	2.78	3.23	1	18
19	2	200	8.40	3.56	1	19
20	3	275	25.4	6.15	0	20
21	3	310	28.6	5.46	1	21
22	2	220	9.24	3.23	2	22
23	2	320	13.4	2.22	0	23
24	1	132	3.67	2.45	2	24
25	2	205	8.61	3.47	0	25
26	2	200	8.40	3.56	1	26
27	1	190	5.29	1.70	3	27
28	2	170	7.14	4.18	1	28
29	1	189	5.51	1.63	2	29
30	1	155	4.31	2.09	1	30
31	1	200	5.56	1.62	2	31

Tabela 5: Dados para os ramos - Sistema FURNAS

## VII.2

ARVORE DE EXPANSÃO

1..2	1..5	1..9	1..3	1..7-----	VIABEL	1
				2..4-----	VIABEL	2
				3..9-----	VIABEL	3
			2..7	1..4-----	VIABEL	4
			3..4-----	VIABEL	5	
			4..9	1..4-----	VIABEL ELIM.	
		2..8	1..3	1..7-----	VIABEL	6
				2..9	1..7-----	VIABEL ELIM.
				2..9-----	VIABEL ELIM.	
				3..4-----	VIABEL	7
			2..4-----	VIABEL	8	
			3..7	1..4-----	VIABEL ELIM.	
			4..9	1..7	1..4-----	VIABEL ELIM.
				2..4-----	VIABEL ELIM.	
				3..9	1..4-----	VIABEL ELIM.
	2..1	1..9	1..3	1..7-----	VIABEL	9
				2..4-----	VIABEL +	
				3..9-----	VIABEL	10
			2..7-----	SEM SUCESSOR		
			3..4-----	VIABEL +		
			4..9-----	SEM SUCESSQR		
		2..8	1..3	1..7-----	VIABEL	11
				2..9-----	SEM SUCESSOR	
			2..7-----	SEM SUCESSOR		
			3..9	1..7-----	SEM SUCESSOR	
			3..9-----	SEM SUCESSOR		
			4..8-----	SEM SUCESSOR		
2..5		1..9-----		SEM SUCESSOR		
		2..8-----		SEM SUCESSOR		
3..1		1..9-----		SEM SUCESSOR		
		2..8-----		SEM SUCESSOR		
4..9-----				SEM SUCESSOR		

LISTA DE SUCESSORAS

ANO 1 55 CONTINGENCIAS

	1	2	3	4	5	6	7	8	9	REG	CUSTO
1	0	1	1	0	1	0	1	1	0	6	30.74
2	0	1	1	0	1	0	1	0	1	1	32.91
3	0	1	1	0	1	0	0	0	2	3	33.67
4	1	1	1	0	0	0	1	1	0	11	34.48
5	0	1	0	1	1	0	0	1	0	8	35.77
6	1	1	1	0	0	0	1	0	1	9	36.65

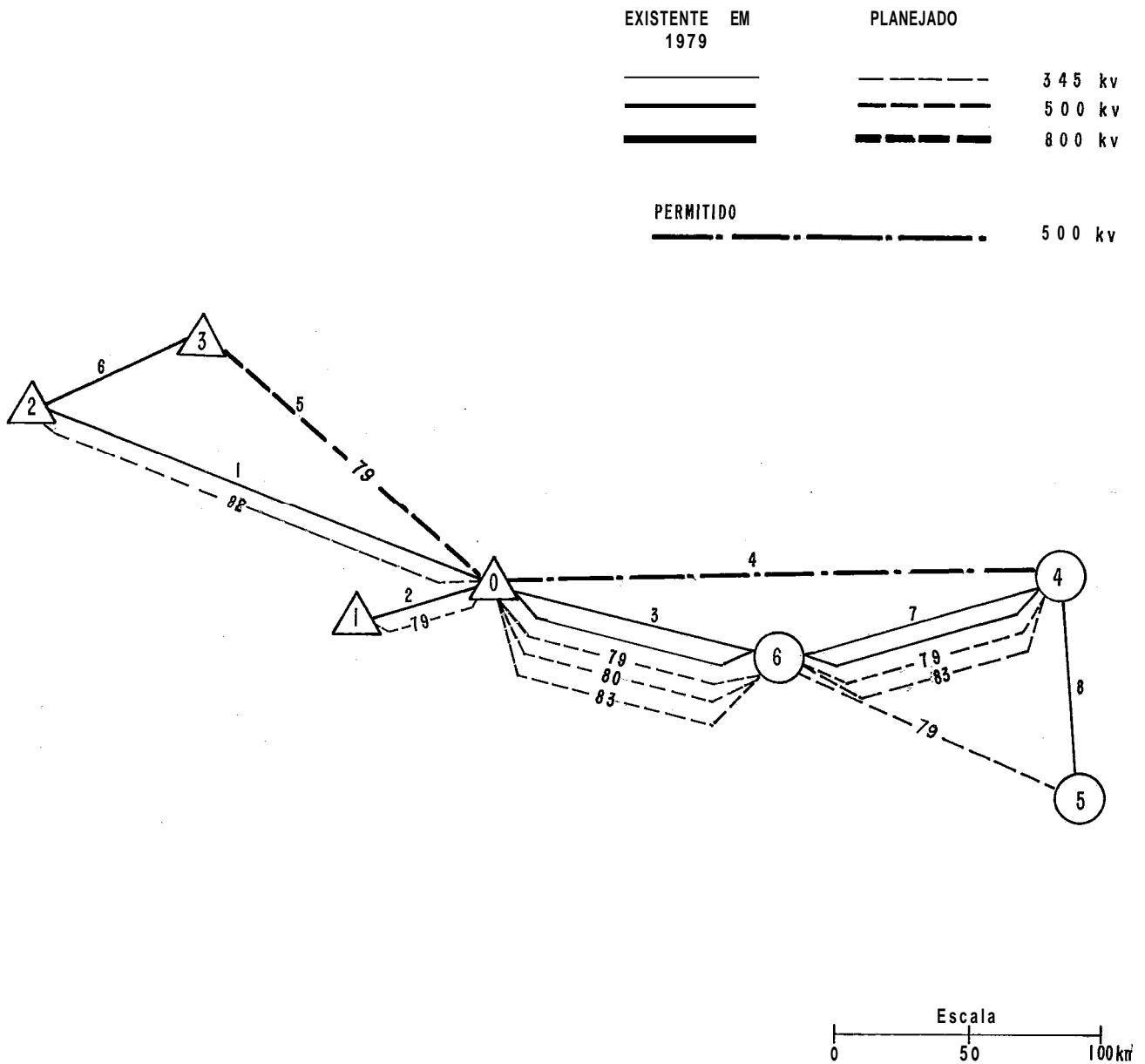


Figura VII.1: Sistema CEMIG - Estratégia ótima

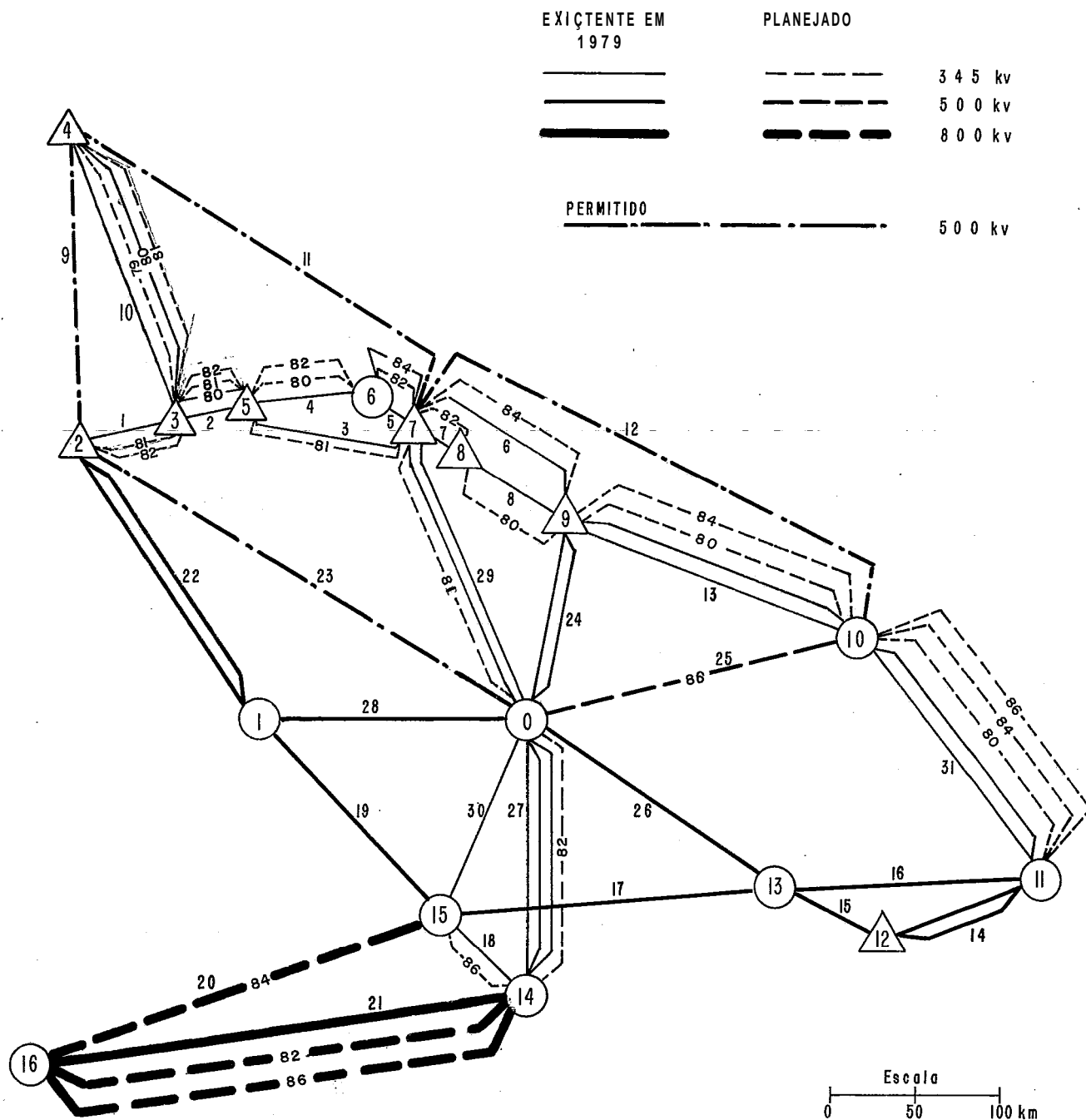


Figura VII.2: Sistema FURNAS - Estratégia incremental

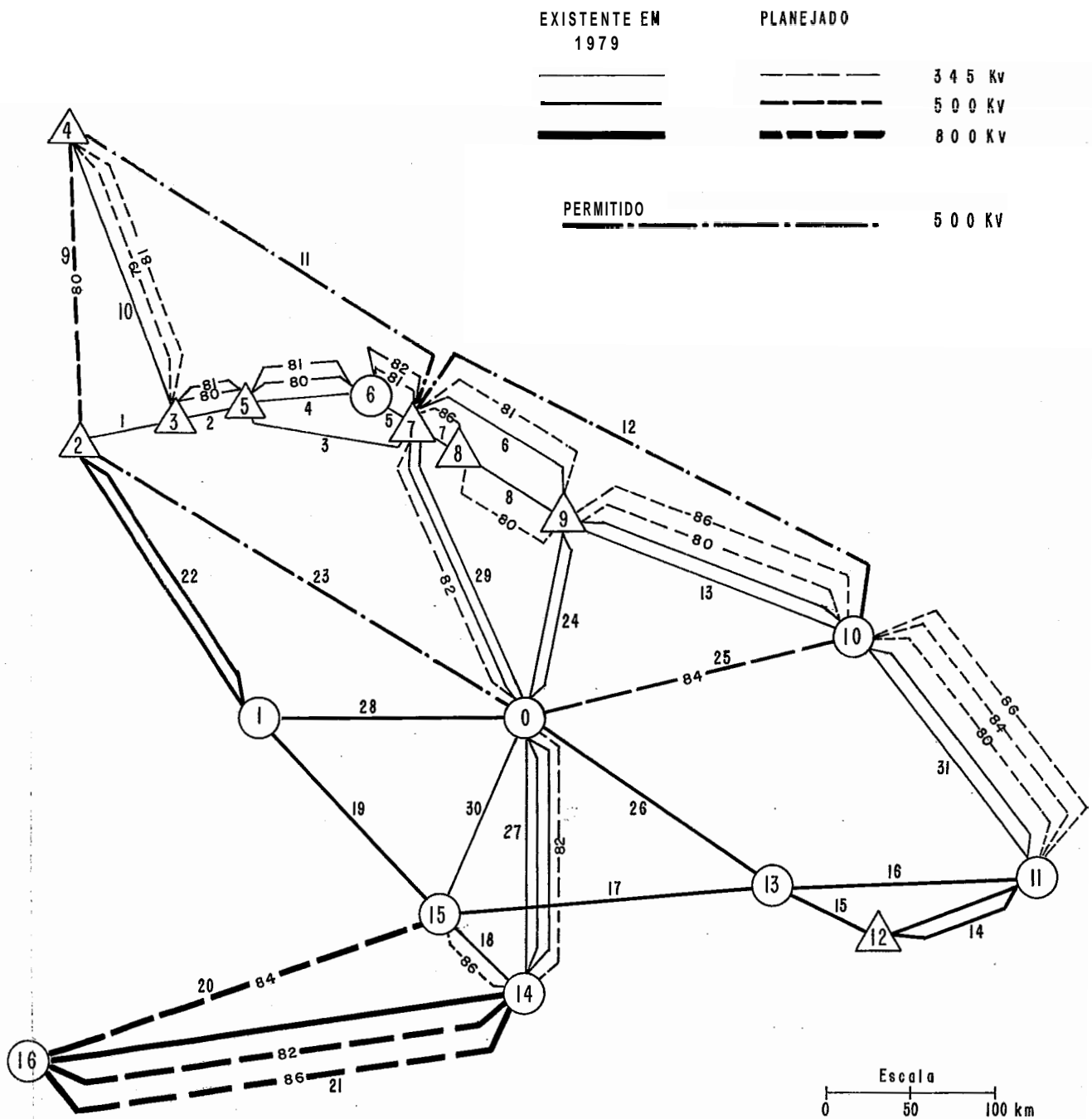


Figura VII.3: Sistema FURNAS - Estratégia Ótima

### Tratamento de sistemas complexos

$h$  medida que aumenta o porte do problema, torna-se necessária ao usuário uma interação maior com o algoritmo. Uma prática que produziu bons resultados consiste em obter inicialmente uma política incremental (ver V.3.57), fazendo poucas restrições ao porte da árvore gerada pelo algoritmo de expansão. Baseado nesses resultados, pode-se resolver completamente o problema, ou rodar algumas vezes o programa, jogando com o formato dos conjuntos  $\delta_i(\cdot, \cdot)$ .

Isto é feito restringindo os conjuntos de estados sucessores permissíveis para alguns ramos, fazendo um bom planejamento de uma região do sistema; relaxam-se em seguida as restrições para aqueles ramos, fixando o desenvolvimento da região planejada. Este pode ser também o procedimento adotado para o tratamento das economias de escala, quando houver vários tipos de linhas permissíveis em cada ramo: a partir de um projeto com linhas de pequena capacidade, fazem-se várias aplicações do programa, permitindo em cada aplicação que um ramo receba linhas de vários tipos, fixando o desenvolvimento dos demais. Esse procedimento corresponde ao "Discrete Nominal" desenvolvido em [10].

## CAPÍTULO VIII

### CONCLUSÃO

As técnicas desenvolvidas nessa tese destinam-se à resolução de problemas relacionados a sistemas de grande porte.

Tais sistemas costumam ser bastante estruturados, o que pode fornecer elementos heurísticos capazes de abreviar seu tratamento.

A Programação Heurística procura isolar a aplicação de certos procedimentos heurísticos, dando-lhes um tratamento formal que permite prever os tipos de erros que podem surgir devido à sua aplicação.

O algoritmo de programação heurística proposto neste trabalho procura obter soluções sub-ótimas para problemas de busca em grafos. Sua utilização é aconselhável quando for possível obter rapidamente soluções sub-Ótimas extraíndo-se dessas soluções informações capazes de levar a soluções melhores. Obviamente, só tem sentido recorrer-se a essas técnicas quando o porte do problema impedir sua resolução diretamente.

A técnica desenvolvida, bem como qualquer método de busca em grafos, pode ser imediatamente aplicada à resolução de problemas de decisões sequenciais: isto é feito com relação a problemas de planejamento de expansão de sistemas descritos por redes e, especificamente, a sistemas de transmissão de energia elétrica.



## VIII.

Os resultados computacionais obtidos para esses últimos sistemas são bons, e a técnica é suficientemente eficiente para o planejamento de vários sistemas de transmissão existentes no Brasil.

A validade dos resultados obtidos depende do modelo usado na avaliação do desempenho dos sistemas de potência: embora o fluxo de carga linearizado forneça com precisão razoável as defasagens nos ramos da rede de transmissão, falta ainda um estudo de estabilidade que forneça critérios de verificação rápida baseados nessas defasagens.

A aplicação do método proposto ao planejamento de outros tipos de sistemas descritos por redes exigirá certamente o desenvolvimento de novas técnicas, principalmente relacionados à expansão de cada configuração. É importante ter-se em mente que a aplicação de qualquer técnica à resolução de um problema de grande porte é, em si, um problema de grande porte, para o qual a técnica escolhida serve de apoio, aliando-se à intuição e à experiência obtida no contato com os sistemas.

## APÊNDICE A - TEORIA DE GRAFOS

Neste Apêndice listam-se algumas definições e resultados da teoria de grafos, seguindo a referência [18]. Não se procura aqui um rigorismo muito grande e uma apresentação formal de grafos encontra-se em [16].

Um grafo é um par ordenado  $G=(N,M)$ , onde

$N$  é um conjunto, cujos elemtnos são os nós de  $G$

$M$  é uma família,  $M=(r_i)_{i \in I}$ , cujos elementos, os ramos de  $G$  satisfazem

$$r_i \in N \times N, \quad i \in I$$

Um ramo é, portanto, um par  $r=(n_1, n_2)$ , onde  $n_1, n_2 \in N$ .

$n_1$  é a extremidade inicial de  $r$  e  $n_2$  é a extremidade final de  $r$

Observe-se que  $H$  é definido como uma família e portanto, pode ocorrer  $r_j=r_k$ ,  $j \neq k$ , ou seja, pode haver ramos repetidos.

- 2 Um grafo  $(N,M)$  é uma rede se não tiver ramos do tipo  $(n,n)$ , isto é, se não tiver laços. Um grafo é finito se  $N,M$  são finitos.

Dois ramos de  $G$  são adjacentes se tiverem ao menos uma extremidade em comum. Dois nós de  $G$  são adjacentes se forem extremidades de um ramo.

Seja  $\bar{n} \in N$ . Definem-se os seguintes conjuntos:

- 3  $w^+(\bar{n}) = \{r \in M \mid r = (\bar{n}, n), n \in N\}$ , conjunto dos ramos incidentes a  $\bar{n}$  para o exterior, ou ramos emergentes de  $\bar{n}$ .

## APÊND. A

- 4  $w^-(\bar{n}) = \{ r \in M \mid r = (n, n), n \in N \}$ , conjunto dos ramos incidentes a  $\bar{n}$  para o interior, ou ramos imergentes a  $\bar{n}$ .

$w(\bar{n}) = w^+(\bar{n}) \cup w^-(\bar{n})$ , conjunto de ramos incidentes a  $\bar{n}$ .

- 5 Um caminho é uma sequência de ramos tais que a extremidade final de cada ramo (com exceção do Último) coincide com a extremidade inicial do seguinte, por exemplo,

$$((n_1, n_2), (n_2, n_3), (n_3, n_4)).$$

O comprimento de um caminho é o número de ramos que o compõem. Um circuito é um caminho em que coincidem a extremidade inicial do primeiro ramo e a extremidade final do Último.

- 6 Uma cadeia (ou caminho não orientado) é uma sequência de ramos  $(r_1, r_2, \dots, r_p)$  tais que cada ramo intermediário é ligado ao anterior por uma extremidade e ao seguinte pela outra extremidade. Se as extremidades de  $r_1$  e  $r_p$  não ligadas a  $r_2, r_{p-1}$  coincidirem, a cadeia recebe o nome de ciclo.

- 7 Um grafo é conexo se quaisquer dois nós são ligados por meio de alguma cadeia.

- 8 Seja  $G=(N, M)$  um grafo. Um grafo  $(N, M')$  com  $M' \subset M$  é um grafo parcial de  $G$ .  $(N', M')$  é um subgrafo de  $G$  se  $N' \subset N$ ,  $M' \subset M$  e todos os ramos de  $M$  cujas extremidades estão em  $N'$ , estão também em  $M'$ , isto é,  $M' = \{ (n, n') \in M \mid n, n' \in N' \}$ . Um subgrafo parcial de  $G$  é um grafo parcial de um subgrafo de  $G$ , isto é,  $(N', M')$  é um subgrafo parcial de  $(N, M)$  se  $(N', M')$  é um grafo e  $N \subset N'$ ,  $M \subset M'$ . As definições equivalentes para redes levam a sub-redes, redes parciais e sub-redes parciais.

## APÊND. A

- 9 Um nó  $\bar{n}$  é um centro do grafo  $(N, M)$  se qualquer nó de  $N$  pode ser atingido por um caminho cuja extremidade inicial é  $\bar{n}$ .
- 10 Uma árvore é um grafo conexo sem ciclos. Em uma árvore, cada par de nós pode ser ligado por uma Única cadeia. Uma árvore munida de um centro é uma arborecência. Em uma arborecência, cada nó pode ser ligado ao centro por um único caminho. Note-se que uma árvore não pode admitir ramos múltiplos.
- 11 Se um grafo  $G=(N, M)$  não tem ramos múltiplos, isto é, se  $(\forall r, r' \in M) \quad r \neq r'$ , então o grafo é perfeitamente caracterizado pelo par  $(N, \Gamma)$ , onde  $\Gamma$  é o operador sucessor de  $G$ , definido por
- $$\forall \bar{n} \in N \quad \Gamma(\bar{n}) = \{n \mid (\bar{n}, n) \in M\}$$
- ou seja,  $\Gamma(\bar{n})$  é o conjunto das extremidades terminais dos ramos e-mergentes de  $\bar{n}$ . Os nós em  $\Gamma(\bar{n})$  são chamados nós sucessores de  $\bar{n}$ . Um grafo sem ramos múltiplos pode ser indiferentemente indica-do por do por  $G=(N, M)$  ou por  $G=(N, \Gamma)$ , de acordo com a conveniê-n-cia.
- Em um grafo sem ramos múltiplos, um caminho  $C$  pode ser representado pela família de nós visitadas por  $C$ :
- $$C = (n_1, n_2, \dots, n_k) \equiv ((n_1, n_2), (n_2, n_3), \dots, (n_{k-1}, n_k))$$
- 12 Se  $G = (N, \Gamma)$  é um grafo com centro  $n_0$ ,  $G$  fica to-talmente especificado por  $n_0$  e  $i'$ : De fato,  $N = \bigcup_i \{\Gamma^k(n_0) \mid k=0, 1, \dots\}$

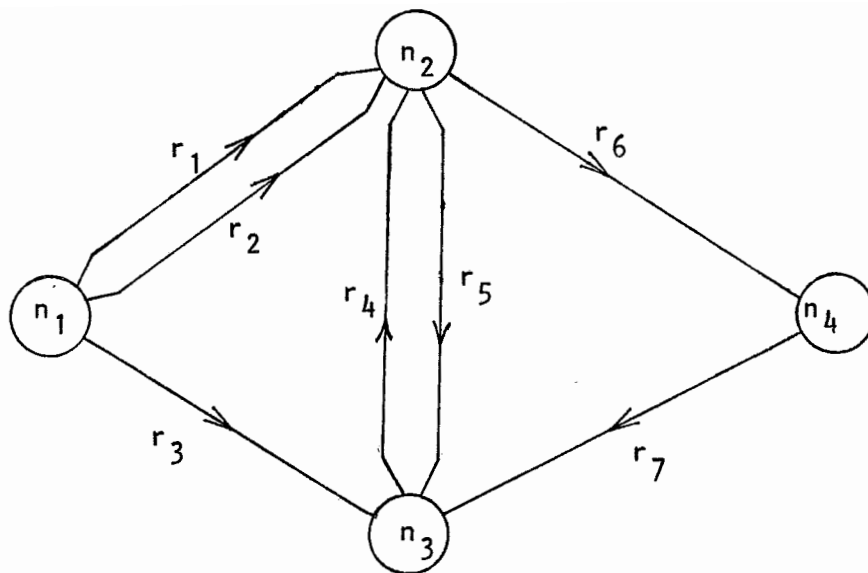


Fig. A.1 - Exemplo de Grafo

$$N = \{n_1, n_2, n_3, n_4\}$$

$$M = (r_i)_{i=1,2,\dots,7} = ((n_1, n_2), (n_1, n_2), (n_1, n_3), \dots, (n_4, n_3))$$

$(N, M)$  é uma rede finita, com um ramo duplo,  $r_1 = r_2$

Exemplos:

Ramos adjacentes:  $r_1, r_2$  ou  $r_6, r_7$ , etc

Ramos emergentes de  $n_2$ :  $w^+(n_2) = \{r_5, r_6\}$

Ramos imergentes a  $n_2$ :  $w^-(n_2) = \{r_1, r_2, r_4\}$

Ramos adjacentes a  $n_2$ :  $w(n_2) = \{r_1, r_2, r_4, r_5, r_6\}$

Caminho:  $(r_3, r_4, r_6)$

Circuitos:  $(r_6, r_7, r_4)$  ou  $(r_5, r_4)$  ou  $(r_5, r_4, r_6, r_7, r_4)$

Ciclos:  $(r_1, r_2)$  ou  $(r_5, r_3, r_2)$  ou qualquer circuito acima

Grafo parcial:  $G' = (N, M')$  com  $M' = (r_2, r_3, r_6, r_7)$

Subgrafo:  $G' = (N', M')$  com  $N' = (n_1, n_2, n_4)$ ,  $M' = (r_1, r_2, r_6)$

Subgrafo parcial:  $G' = (N', M')$  com  $N' = (n_1, n_2, n_4)$ ,  $M' = (r_2, r_6)$

Operador sucessor:  $\Gamma(n_1) = \{n_2, n_3\}$ ,  $\Gamma(n_4) = \{n_3\}$

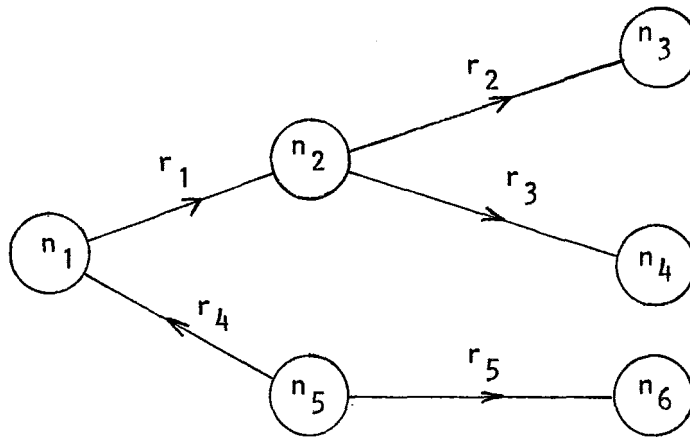


Fig. A.2 - Árvore

(Trata-se, na realidade, de uma arborescência com centro  $n_5$ . De fato, qualquer nó pode ser atingido por um Único caminho partindo de  $n_5$ . Este não seria o caso se invertêssemos a orientação de  $r_3$ ).

### Fluxos e Tensões

- 13 Seja  $(N, M)$  uma rede finita, com  $M = \{r_1, r_2, \dots, r_m\}$ ,  $N = \{n_1, n_2, \dots, n_n\}$ . Seja  $P \in \mathbb{R}^n$  um vetor de injeções de fluxo, que associa a cada nó  $j$  uma injeção  $P_j$ . Um vetor  $f \in \mathbb{R}^m$  é um fluxo para a rede  $(N, M)$  com fontes e sumidouros se

$$P_j + \sum_{i \in W^-(n_j)} f_i = \sum_{i \in W^+(n_j)} f_i \quad j=1, 2, \dots, n$$

A expressão (13) corresponde à primeira lei de Kirschhoff.

## APÊND. A

14 Um vetor  $\phi \in R^m$  é uma tensão se existe um vetor  $\theta \in R^n$  tal que para todo ramo  $r_i = (n_p, n_q)$

$$\phi_i = \theta_q - \theta_p$$

O vetor  $\theta$  é chamado potencial associado à tensão  $\phi$ .

Matriz de incidência

Define-se matriz de incidência de um grafo  $(N, M)$  de  $n$  nós e  $m$  ramos como uma matriz  $M(n \times m)$  tal que

$$15 \quad M_{ji} = \begin{cases} 1 & \text{se } r_i \in w^+(n_j), \text{ ou } r_i \text{ emerge de } n_j \\ -1 & \text{se } r_i \in w^-(n_j), \text{ ou } r_i \text{ imerge em } n_j \\ 0 & \text{se } r_i \notin w(n_j), \text{ ou } r_i \text{ não é adjacente a } n_j \end{cases}$$

Exemplo: a matriz de incidência para a rede da fig.A.1

é :

$$M = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & -1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 1 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \end{matrix}$$

Observe-se que cada linha de  $M$  descreve a composição do conjunto de ramos adjacentes a um nó. Cada coluna tem sempre dois elementos não nulos, que indicam as extremidades de um nó.

Análise matricial de fluxos e tensões

Sejam  $f \in R^m$  um fluxo,  $\theta \in R^n$  um potencial e  $\phi \in R^m$  uma tensão. São válidas as seguintes expressões para a re-

de (13) :

$$P = M f$$

$$\phi = -M^T \theta$$



APÊNDICE B - O ALGORITMO A\*

Faz-se a seguir um breve resumo sobre o algoritmo  $A^*$ , destinado à busca de caminhos de custo mínimo em grafos. O algoritmo utiliza uma estimativa, associada a cada nó do grafo, do custo de um caminho Ótimo iniciado no nó considerado.

Seu enunciado original encontra-se em [6], e a versão transcrita abaixo foi retirada da referência [2], onde também se encontra um tratamento extenso de diversos algoritmos de busca em grafos.

1 Considere-se neste apêndice um grafo simples  $H=(N,\Gamma)$ , onde  $\Gamma : N \times N \rightarrow \mathcal{P}(N)$ .  $H$  pode ser um grafo infinito, mas o conjunto de sucessores  $\Gamma(n)$  de qualquer nó  $n \in N$  deve ser finito.

2 Sendo  $M$  o conjunto de ramos de  $H$  (Ver Apêndice A), considera-se conhecida uma função  $c : M \rightarrow \mathbb{R}$ , que associa a cada ramo  $r=(n_1, n_2)$  o custo  $c(r) = c(n_1, n_2)$ . A um caminho  $G = (n_0, n_1, \dots, n_p)$  em  $H$  associa-se o custo

$$c(G) = \sum_{i=1}^p c(n_{i-1}, n_i)$$

3 Procura-se, dados um nó inicial  $s \in M$  e um conjunto alvo  $T \subset N$ , encontrar um caminho de  $s$  para  $T$ , cujo custo é mínimo entre todos os tais caminhos.

4 Dados  $s, n_1, n_2 \in N$ ,  $A, B, T \subset N$ , definem-se:

$h(n_1, n_2)$  : custo de um caminho Ótimo entre  $n_1$  e  $n_2$

$h(A, B) \triangleq \min\{h(n_1, n_2) \mid n_1 \in A, n_2 \in B\}$

## APÊND. B

$$h(n_1) \triangleq h(n_1, T)$$

$$h(A) \triangleq h(A, T)$$

$$g(n_1) \triangleq h(s, n_1)$$

Com essas definições, o problema de busca consiste em encontrar um caminho  $G$  entre  $s$  e  $T$  tal que

$$c(G) = h(s) .$$

O algoritmo  $A^*$  emprega alguma informação heurística a respeito da estrutura do grafo, através de uma função avaliação. Essa função associa a cada nó  $n$  de  $H$  uma estimativa sobre o valor de um caminho de  $s$  para  $T$  com custo mínimo entre todos os caminhos que passam por  $n$  .

Esta estimativa  $\hat{f}$  consiste de duas parcelas ,  $\hat{f}(n) = \hat{g}(n) + \hat{h}(n)$  , onde  $\hat{g}(n)$  é calculada pelo algoritmo e comsistem no caminho de menor custo até  $n$  encontrado pelo algoritmo até a iteração considerada.  $\hat{h}(n)$  é uma estimativa sobre o ca-minho entre  $n$  e  $T$  , exemplificada pela distância "via aérea" para o problema de atravessar um terreno com obstáculos.

O algoritmo manipula duas listas, Aberto e Fechado. Os nós são colocados em Aberto ao serem gerados pelo algoritmo.  $A^*$  escolhe em cada iteração um nó correspondente ao menor valor de  $\hat{f}$  presente em Aberto. Este nó é expandido (isto é, obtem-se a des - crição de seus sucessores) e é transferido para Fechado. Os novos nós gerados entram em Aberto e continua-se a busca. Quando um nó

alvo for escolhido pelo algoritmo, termina-se o algoritmo.

## 5 Algoritmo: $A^*$

Passo 1 : Coloque o nó inicial  $s$  em uma lista chamada Aberto e calcule  $\hat{f}(s)$

Passo 2 : Se Aberto estiver vazia, pare (com insucesso)  
Senão, continue

Passo 3 : Retire de Aberto o nó cujo valor de  $\hat{f}$  for menor e introduza-o em uma lista chamada Fechado. Seja  $n$  este nó (resolva empates arbitrariamente, mas sempre a favor de qualquer nó do alvo).

Passo 4 : Se  $n \in T$ , termine, obtendo o caminho solução por meio dos indicadores. Senão, continue.

Passo 5 : Expanda  $n$ , gerando  $\Gamma(n)$  (Se  $\Gamma(n) = \emptyset$ , vá para o passo 2). Para cada sucessor  $n_i$ , calcule  $\hat{f}(n_i)$  por meio de (6).

Passo 6 : Associe aos sucessores que ainda não se encontram em Aberto nem em Fechado os valores de  $\hat{f}$  calculados. Introduza esses nós em Aberto e dirija apontadores a  $n$ .

Passo 7 : Associe aos sucessores que já estavam em Aberto ou Fechado os menores entre os valores de  $\hat{f}$  calculados agora e seus valores prévios.  
Transfira para Aberto aqueles sucessores que estão em Fechado e tiveram os valores de  $\hat{f}$  rebaixados e redirija para  $n$  os apontadores de todos os nós cujos valores de  $\hat{f}$  foram rebaixados.

Passo 8 : Vã para o passo 2 .

6 Utiliza-se  $\hat{f}(n) = \hat{g}(n) + \hat{h}(n)$  , onde

$\hat{g}(s) = 0$  e, no passo 5, calcula-se

$$\hat{g}(n_i) = \hat{g}(n) + c(n, n_i)$$

Um algoritmo de busca em grafos é dito admissível quando sua aplicação fornece garantidamente um caminho de custo mínimo, desde que exista tal caminho.

7 Teorema : Se  $\hat{h}(n) \leq h(n)$  para qualquer nó  $n \in N$  , e se todos os ramos de  $H$  tem custos maiores do que uma constante positiva  $\delta$  , então  $A^*$  é admissível.

A demonstração do teorema (7) encontra-se na referência [2], juntamente com um estudo detalhado das vantagens obtidas pelo uso de heurística.

A demonstração de admissibilidade é feita para 6-grafos, mas pode ser facilmente refeita para grafos finitos em que não existem circuitos de custo negativo.

8 De fato, essas condições garantem que  $h(n, n') > -\infty$  para qualquer par de nós em  $N$  , uma vez que a cada circuito  $G$  entre  $n$  e  $n'$  corresponde um caminho  $\bar{G}$  sem circuitos tal que  $c(\bar{G}) \leq c(G)$  . Além disso, um nó  $n$  somente é recolocado em Aberto quando seu custo  $\hat{g}(n)$  é rebaixado, correspondendo a um novo caminho entre  $s$  e  $n$  . O fato acima e o fato de existir um número finito de caminhos sem circuitos em um grafo finito mostram que  $A^*$  sempre termina. O restante da demonstração de admissibi-

## APÊND. B

lidade é idêntico ao apresentado em [2].

## REFERÊNCIAS

1. Bellman R. e Dreyfus S. - Applied Dynamic Programming, Princeton, N.J., Princeton University Press, 1962.
2. Nilsson, N., - Problem-Solving Methods in Artificial Intelligence, McGraw-Hill, 1971.
3. Dijkstra, E.W. - A Note on Two Problems in Connection with Graphs, Numerische Mathematik 1 : 269-271 , 1959.
4. Dreyfus, S. - An appraisal of Some Shortest Path Algorithms , Operations Research 17 , nº 3 : 395-412 , Maio-Junho, 1969.
5. Lawler, E. e Wood D. - Branch and Bound Methods: A Survey, Operations Research 14 , Nº 4 : 699-719, Julho-Agosto, 1966.
6. Hart, P. , Nilsson, N. e Raphael, B. - A Formal Basis for the Heuristic Determination of Minimum Cost Paths, IEEE Trans. Syst. Sc. Cybernetics 4, Nº 2 : 100-107, Julho, 1968.
7. Pohl, I. - Bi-Directional and Heuristic Search in Path Problems, Tese de Doutorado, Computer Sciences Dept., Stanford Univers., Stanford, 1969.
8. Michie, D. - Heuristic Search, The Computer Journal, 14 Nº1: 96-102, 1971 .
9. Puntel, W.R. e outros - An Automated Method for Long-Range Planning of Transmission Networks, PICA Conference, Chicago, Abril/1973.

10. Kaltenbach, J.C. - Problems Relating to the Optimal Planning of Network Capacities, Tese de Doutorado, Dept. of Industrial Engineering, Stanford University, Stanford, 1969.
11. Gonzaga C. , Persiano R. , Carneiro S. e Brito S. - Optimal Planning of the Expansion of a Power Transmission System , IFAC/IFORS Conference on Systems Approaches to Developing Countries , Alger, Maio/1973.
12. Peschon, J. e outros - Power System Planning and Reliability , Relatório Final, SRI Project 6619, Stanford Research Institute, Menlo Park, Junho/1968.
13. Rienstra F. e Clason, R.J. - A Comparison Between Long Term and Short Term Optimization for the Expansion of an Electricity Production System, 4<sup>th</sup> Power Systems Computation Conference, Grenoble, Setembro/1972.
14. Doulliez, P. - Optimal Capacity Planning of Multidimensional Networks , Tese de Doutorado, Université Catholique de Louvain, Bélgica, 1970.
15. Dusonchet, Y. , El-Abiad, A. - Transmission Planning Using Discrete Dynamic Optimizing, IEEE Summer Power Conference , S. Francisco, Julho/1972.
16. Berge, C. - Graphes et Hypergraphes, Dunod, Paris, 1970.
17. Hu, T.C. - Integer Programming and Network Flows , Addison Wesley, 1969.

18. Berge,C. , Gho ila-Houri,A. - Programmes, Jeux et Réseaux de Transport, Dunod, Paris, 1962.
19. Ford,L. e Fulkerson - Flows in Networks , Princeton University Press, Princeton 1962,
20. Meier Jr.,W. - An Optimization Method for Branching Multistage Water Resource Systems , Water Resources Research,3 N? 3 : 645-652 , 1967.
21. Bertele,U. e Brioschi,F.- Nonserial Dynamic Programming,A.Press/72.
22. Morin,T. e Esogbue,A. - Some Efficient Dynamic Programming Algorithms for the Optimal Sequencing and Scheduling of Water Supply Projects, Water Resources Research, 7 , N?3: 479-484, Junho/1971.
23. Butcher,W. , Haimes,Y e Hall,W. - Dynamic Programming of the Optimal Sequencing of Water Supply Projects, Water Resources Research, 5 N?6: 1196-1204, Dezembro/1969.
24. Análise de Benefícios: Sistema Furnas , Nota DPE.0-73, Departamento de Planejamento Energético, Furnas-Centraís Elétricas S.A., Março/1972.
25. Norris,T.E. - Economic Comparisons in Planning for Electricity Supply, Proceedings IEEE, Vo1.117, N?3: 593-605, Março/1970.
26. Bills,G.W. - Electric Power System Transient Stability Control for the 70's , 1970 Purdue Symposium on Power Systems.
27. Dy Liacco,J. , Wirts,B. e Wheeler,D - Automation of the CEI



System for Security, 1971, PICA Conference .

28. Gonzaga.C. e Brito S. - Planejamento Ótimo da Expansão de Sistemas de Transmissão de Potência, apresentado no Seminário Nacional de Produção e Transmissão de Energia Elétrica, Belo Horizonte, Brasil, Setembro/1973.
29. Kaltenbach,J. , Peschon,J. e Gehring,E. - A Mathematical Optimization Technique for the Expansion of Electric Power Transmission Systems, IEEE Trans. on Power Apparatus and Systems , PAS-89, N° 1: 113-119, Janeiro/1970.
30. Carpentier,M.J. - Contribution à l'Étude du Dispatching Économique, Buletin de la Société Française des Electriciens, Agosto/1962.
31. Federal Power Commission Advisory Committee Report n° 6: Criteria for AC Transmission System Performance, National Power Survey, Federal Power Commission, U.S. Government Printing Office : 43-47, 1964.
32. Gonzaga.C., Persiano R. - 2º Relatório Técnico: Projeto COPPETEC P. 181/71, Estudos Elétricos, Rio de Janeiro, Outubro/1973.
33. Nemhauser, G. - A Generalized Permanent Label Setting Algorithm for the Shortest Path Between Specified Nodes, Journal of Mathematical Analysis and Applications, 38 : 328-334 , 1972.
34. Katayama,H. - Estudo de Problemas Relacionados à Busca Heurística Bi-direcional em Grafos : Tese de Mestrado, COPPE, a ser apresentada em novembro/1973.

35. Golomb, S. e Baumert, L. - Backtrack Programming, J.ACM, 12 , N? 4: 516-524, Outubro/1965.